



SWsoft, Inc.

Virtuozzo

User's Guide

Version 3.0



(c) 1999-2006

ISBN: N/A
SWsoft Inc
13755 Sunrise Valley Drive
Suite 325
Herndon, VA 20171
USA
Tel: +1 (703) 815 5670
Fax: +1 (703) 815 5675

Copyright © 1999-2006 by SWsoft, Inc. All rights reserved
Distribution of this work or derivative of this work in any form is prohibited unless prior written permission is obtained from the copyright holder.
Linux is a registered trademark of Linus Torvalds.
Virtuozzo, Plesk, HSPcomplete, and corresponding logos are trademarks of SWsoft, Inc.
Red Hat is a registered trademark of Red Hat Software, Inc.
SUSE is a registered trademark of Novell, Inc.
Solaris is a registered trademark of Sun Microsystems, Inc.
X Window System is a registered trademark of X Consortium, Inc.
UNIX is a registered trademark of The Open Group.
Intel, Pentium, and Celeron are registered trademarks of Intel Corporation.
IBM DB2 is a registered trademark of International Business Machines Corp.
SSH and Secure Shell are trademarks of SSH Communications Security, Inc.
MegaRAID is a registered trademark of American Megatrends, Inc.
PowerEdge is a trademark of Dell Computer Corporation.

Contents

Preface	10
About This Guide	10
Who Should Read This Guide	11
Organization of This Guide	12
Documentation Conventions.....	13
Typographical Conventions.....	13
Shell Prompts in Command Examples	14
General Conventions	14
Feedback	14
Virtuozzo Philosophy	15
About Virtuozzo Software	15
What is Virtuozzo.....	15
Virtuozzo Applications.....	16
Distinctive Features of Virtuozzo	16
OS Virtualization.....	17
Network Virtualization	17
Virtuozzo File System (VZFS).....	18
Templates	18
Resource Management	19
Main Principles of Virtuozzo Operation.....	20
Basics of Virtuozzo Technology	20
Understanding Templates	21
Virtuozzo Configuration.....	23
Understanding Licensing.....	24
VZMC Overview.....	24
VZCC Overview	28
VZPP Overview.....	29
Hardware Node Availability Considerations	30
Operations on Virtual Private Servers	31
Creating New Virtual Private Server	31
Before You Begin.....	32
Choosing Virtual Private Server ID.....	33
Creating VPS on Basis of OS Standard Template.....	34
Creating VPS on Basis of OS EZ Template	37
Creating Virtual Private Servers in VZMC	41
Configuring Virtual Private Server.....	44
Setting Startup Parameters.....	44
Setting Network Parameters	45
Setting root Password for VPS	46
Starting, Stopping, Restarting, and Querying Status of Virtual Private Server	47
Listing Virtual Private Servers.....	49
Migrating Virtual Private Server	51
Standard Migration.....	51
Zero-Downtime Migration	54
Backing Up and Restoring Virtual Private Servers.....	56
Running Backup Utilities on Backup Node.....	57

Managing Backups in VZMC.....	59
Reinstalling Virtual Private Server	67
Customizing VPS Reinstallation	69
Deleting Virtual Private Server.....	71
Disabling VPS	73
Running Commands in Virtual Private Server	74

Managing Templates 75

Template Lifecycle	76
Listing Templates	77
Creating and Installing Application Templates.....	78
Creating and Installing Template Updates.....	79
Adding Templates and Template Updates to VPSs	81
Removing Templates and Template Updates From VPSs	82
OS Template Updates vs Application Templates	83
Creating OS Template With vzsnapshot.....	84
Creating OS Template	85
Preparing Virtual Private Server for Building OS Templates.....	86
Preparing Packages.....	87
Writing Template Configuration File	87
Building and Testing OS Template	91
Creating OS Template in VZMC.....	92

Managing EZ Templates 93

Understanding EZ Templates	93
Overview	93
EZ Template Directory Structure	95
EZ Template Lifecycle	98
Preparing OS EZ Template for VPS Creation	98
Creating VPS on Basis of OS EZ Template	99
Choosing OS EZ Template.....	100
Creating Virtual Private Server	102
Installing Application EZ Templates on Hardware Node.....	103
Adding Application EZ Templates to VPS.....	104
Listing EZ Templates.....	107
Updating EZ Templates in VPS.....	108
Removing Application EZ Template From VPS	109

Managing Resources 110

What are Resource Control Parameters?	110
Managing Disk Quotas	111
What are Disk Quotas?	112
Disk Quota Parameters	112
Turning On and Off Per-VPS Disk Quotas	113
Setting Up Per-VPS Disk Quota Parameters	115
Turning On and Off Second-Level Quotas for Virtual Private Server	118
Setting Up Second-Level Disk Quota Parameters	120
Checking Quota Status	123
Cleaning Up Virtual Private Servers	124
Managing CPU Share	129
Managing Network Accounting and Bandwidth.....	131
Network Traffic Parameters	131
Configuring Network Classes.....	132
Viewing Network Traffic Statistics	134
Turning On and Off Network Bandwidth Management	135

Configuring Network Bandwidth Management for VPS.....	137
Managing System Parameters.....	140
Monitoring System Resources Consumption	143
Monitoring Memory Consumption.....	145
Managing VPS Resources Configuration	146
Splitting Hardware Node Into Equal Pieces	148
Scaling Virtual Private Server Configuration.....	149
Validating Virtual Private Server Configuration	151
Real-Time Monitoring in Virtuozzo	153
Monitoring Resources in Text Console	154
Monitoring Resources in VZMC	156
Using Charts Representation	157
Using Table Representation.....	164
Subscribing for Alerts.....	165
Subscribing for VZMC Alerts	165
Subscribing for vzrmond Alerts	166
Managing Services and Processes	167
What Are Services and Processes	168
Main Operations on Services and Processes	169
Managing Processes and Services	170
Viewing Active Processes and Services	171
Monitoring Processes in Real Time.....	173
Changing Services Mode.....	176
Determining VPS Identifier by Process ID.....	177
Starting, Stopping, and Restarting Services.....	178
Name-Based Hosting for Virtuozzo	180
Name-Based Hosting Overview	181
Basic Configuration	183
Advanced Configuration.....	184
Defining Global Protocol Parameters.....	185
Setting Up VPS Permissions	186
Example.....	186
Setting Up SMTP Disk Quota	187
Learning Current Name-Based Hosting Status	188
Keeping Your Virtuozzo System Up-to-Date	189
Updating Host OS Software	189
Using up2date.....	190
Using yum	190
Using yast.....	190
Updating Virtuozzo Software	191
Updating Software Inside Virtual Private Servers	191
Customizing VZCC/VZPP	192
Overview	193
Common Tasks	194
Adding VZCC/VZPP Links.....	194
Setting Auto Logout Time	196
Setting Maximal Backups Number.....	197

Adjusting Traffic Representations	198
VZCC/VZPP Architecture	199
Customizing Individual Pages	201
Modifying Existing Page	201
Adding New Page	203
Customizing VZCC/VZPP Menu	206
Customizing VZCC/VZPP Layout	208
Adding Skins to VZCC/VZPP	208
Setting Up VZCC/VZPP Demo Mode for Potential Customers	210
Creating "Demo" User	210
Customizing VZCC/VZPP for Using in Demo Mode	212
Managing Demo Mode for VZPP With Plesk Integrated	216
Importing Plesk Language Packs Into VZPP	219
Advanced Tasks	220
Configuring Capabilities	220
Creating VZFS Symlinks Inside VPS	221
Available Capabilities for Virtual Private Server	222
Migrating Physical Server to VPS	224
Migration Overview	224
Migration Steps	224
Migration Requirements	226
Migration Restrictions	228
Migrating Physical Server to VPS in Command Line	229
Migrating Physical Server to VPS in VZMC	235
Migrating VPS to Physical Server	244
Migration Steps	245
Migration Requirements	246
Migrating VPS to Physical Server	246
Changing System Time From VPS	248
Obtaining Hardware Node ID From Inside Virtual Private Server	249
Accessing Devices From Inside Virtual Private Server	250
Moving Network Adapter to Virtual Private Server	252
Enabling VPN for VPS	253
Loading iptables Modules	253
Loading iptables Modules to Hardware Node	254
Loading iptables Modules to Particular VPSs	255
Sharing File System Among Virtual Private Servers	256
Creating Configuration File for New Linux Distribution	257
Rebooting Virtual Private Server	258
Managing Graphical Applications Inside VPS	259
Running Graphical Applications in X Windows	259
Running Graphical Applications via VNC	264
Managing VZCC/VZPP Access Rights	265
Defining VZCC/VZPP User Groups	266
Assigning Access Rights to VZCC/VZPP User Groups	266
Mastering VZMC	270
Managing Node Clusters	270
Creating Cluster	271
Adding Nodes to Cluster	271
Configuring IP Addresses Pool	272
Configuring Virtual Private Server IDs Pool	273
Specifying Cluster Network Settings	274
Checking for Conflicts Among Nodes	275

Managing Offline Services Configuration	276
Viewing Summary Pages	278
Managing Users and Groups	280
Configuring Firewall	282
Managing Mount Points	284
Viewing System and Virtuozone Logs	285
Adjusting Template Configuration Files	286
Managing Files	288
Managing Search Domains	289
Managing Virtual Private Server Servers	289
Checking Virtual Private Server	290
Troubleshooting	291
General Considerations	292
Kernel Troubleshooting	294
Using ALT+SYSRQ Keyboard Sequences	294
Saving and Decoding Kernel Fault (OOPS)	295
Finding Kernel Function That Caused D Process State	296
Problems With VPS Management	296
Failure to Create VPS	296
Failure to Start VPS	297
Failure to Access VPS From Network	298
Failure to Log In to VPS	298
Failure to Back Up VPS in VZMC	299
Failure to Display List of VPS Backups	299
Problems With VPS Operation	300
Timeout When Accessing Remote Hosts	300
Extraneous Backups Visible to VPS in VZPP	300
Problems With Physical Server Migration	300
Failure to Start iptables Modules After Physical Server Migration	300
Miscellaneous Problems	301
Failure to Run vgscan Utility	301
Corrupted Pseudographics in Virtuozone Utilities	301
Getting Technical Support	302
Getting Assistance With Virtuozone Installation	302
Preparing and Sending Questions to Technical Support	302
Submitting Problem Report to Technical Support	303
Establishing Secure Channel to SWsoft Support	305
Glossary	306
Index	309

Table of Figures

Figure 1: Virtuozzo Technology	20
Figure 2: VZMC Network Architecture	26
Figure 3: VZMC Main Window	27
Figure 4: VZMC - Listing Templates	36
Figure 5: VZMC - Listing EZ OS templates	38
Figure 6: VZMC - Creating New Virtual Private Server	41
Figure 7: VZMC - Assigning IP Addresses When Creating Virtual Private Servers	42
Figure 8: VZMC - Checking Newly-Created Virtual Private Server	43
Figure 9: VZMC - Viewing List of VPSs	50
Figure 10: VZMC - Migrating VPSs	53
Figure 11: VZMC - Migrating VPSs	55
Figure 12: VZMC - Backing Up VPSs Wizard	61
Figure 13: VZMC - Restoring VPSs Wizard	63
Figure 14: VZMC - Listing Backups	65
Figure 15: VZMC - Searching for Backups	66
Figure 16: VZMC - Deleting VPS	72
Figure 17: VZMC - Enabling/Disabling VPS	74
Figure 18: VZMC - Listing Templates	77
Figure 19: Creating Virtuozzo Template Update - Specifying Base Directory	80
Figure 20: VZMC - Listing EZ OS templates	100
Figure 21: VZMC - Choosing VPS	105
Figure 22: VZMC - Listing EZ Templates	107
Figure 23: VZMC - Turning First-Level Disk Quota On and Off	114
Figure 24: VZMC - VPS Disk Quota Parameters	115
Figure 25: VZMC - Setting Up VPS Disk Quota	117
Figure 26: VZMC - Turning Second-Level Disk Quota On and Off	119
Figure 27: VZMC - Viewing List of VPS Users	121
Figure 28: VZMC - Setting Up Per-User Disk Quota	122
Figure 29: VZMC - Viewing VPS Disk Quota Statistics	124
Figure 30: VZMC - Configuring Network Classes	133
Figure 31: VZMC - Viewing Network Traffic Statistics	134
Figure 32: VZMC - Setting Up Traffic Shaping Parameters	136
Figure 33: VZMC - Configuring VPS Network Bandwidth	139
Figure 34: VZMC - Validating VPS Configuration	152
Figure 35: VZMC - Adding Monitoring Counters	157
Figure 36: VZMC - Editing Monitor Update Interval	158
Figure 37: VZMC - Adjusting Charts Scale	159
Figure 38: VZMC - Adjusting Charts View	160
Figure 39: VZMC - Accelerating Logs	162
Figure 40: VZMC - Replaying Logs	163
Figure 41: VZMC - Monitoring Traffic Parameters	164
Figure 42: VZMC - Viewing Services	172
Figure 43: VZMC - Monitoring Active Processes	175
Figure 44: VZMC - Managing Processes and Services	178
Figure 45: Name-Based Hosting	181
Figure 46: VZMC - Setting VZCP Auto Logout Time	196
Figure 47: VZCC/VZPP Transformations	200
Figure 48: VZMC - Logging In to Physical Server	235
Figure 49: VZMC - Reviewing Server Configuration	237

Figure 50: VZMC - Customizing Server Migration	238
Figure 51: VZMC - Stopping Services	240
Figure 52: VZMC - Specifying VPS ID and Hostname	241
Figure 53: VZMC - Defining Network Parameters	242
Figure 54: VZMC - Specifying QoS Parameters.....	243
Figure 55: VZMC - Setting DNS Servers.....	274
Figure 56: VZMC - Managing Offline Services Configuration	276
Figure 57: VZMC - Viewing VPS Summary Page.....	278
Figure 58: VZMC - Managing Users and Groups	280
Figure 59: VZMC - Firewall Configuration Dialog	282
Figure 60: VZMC - Managing Mount Points	284
Figure 61: VZMC - Managing Files	288
Figure 62: Submitting Problem Report - Entering name and e-mail address.....	303
Figure 63: Submitting Problem Report - Sending Report to SWsoft.....	304

CHAPTER 1

Preface

In This Chapter

About This Guide.....	10
Who Should Read This Guide.....	11
Organization of This Guide.....	12
Documentation Conventions.....	13
Feedback	14

About This Guide

This guide is meant to provide comprehensive information on Virtuozzo 3.0 – high-end server virtualization software for Linux-based computers. The issues discussed in this guide cover the necessary theoretical conceptions as well as practical aspects of working with Virtuozzo. The guide will familiarize you with the way to create and administer Virtual Private Servers (sometimes also called *Virtual Environments*, or *VEs*) on Virtuozzo-based Hardware Nodes and to employ both graphical and command line interfaces for performing various tasks.

Note: The guide does not familiarize you with the process of installing, configuring, and deploying your Virtuozzo system. Detailed information on all these operations is provided in the *Virtuozzo Installation Guide* shipped with Virtuozzo 3.0.

According to the task-oriented approach, most topics of this guide are devoted to a particular task and the ways to perform it. However, Virtuozzo is equipped with as many as three different tools to perform many administrative tasks:

- 1 The command line interface;
- 2 Virtuozzo Management Console (VZMC) with the graphical user interface;
- 3 Virtuozzo Control Center (VZCC) with web interface.

Besides, there is another tool for managing VPSs - Virtuozzo Power Panels. However, this tool is mainly regarded as a means for individual VPS customers to manage their personal VPSs and is therefore not described in this guide.

VZMC and the command line interface are considered the primary tools for administering Virtuozzo and performing main administrative tasks on Hardware Nodes and in the VPS context. Therefore, when describing the ways to perform this or that task, we have provided the corresponding algorithms only for Virtuozzo Management Console and the command line interface. As to VZPP and VZCC, a web counterpart of VZMC, they are provided with a comprehensive online help system.

Certain Linux administrator's skills are desirable for a person reading the guide. If you foresee any problems with setting up Linux on your computer (for example, related to disk partitioning), you may consult SWsoft Support Team (see **Getting Technical Support** (on page 302) for contacts). In addition, you can obtain some useful information regarding Red Hat OS installation issues from <http://www.redhat.com/docs/manuals/linux/>.

Who Should Read This Guide

The primary audience for this book is anyone responsible for administering one or more systems running Virtuozzo 3.0. To fully understand the guide, you should have strong Linux system administration habits. Attending Linux system administration training courses might be helpful. Still, no more than superficial knowledge of Linux OS is required in order to comprehend the major Virtuozzo notions and learn to perform the basic administrative operations.

Organization of This Guide

Chapter 2, *Virtuozzo Philosophy*, is a must-read chapter that helps you grasp the general principles of Virtuozzo operation. It provides an outline of Virtuozzo architecture, of the way Virtuozzo stores and uses configuration information, of the things you as administrator are supposed to perform, and the common way to perform them. It also lets you understand Virtuozzo licensing policy.

Chapter 3, *Operations on Virtual Private Servers*, covers those operations that you may perform on a VPS as on a single entity: creating and deleting Virtual Private Servers, starting and stopping them, backing up and restoring, etc. You will also learn from this chapter how to migrate VPSs from one Hardware Node to another.

Chapter 4, *Managing Templates*, shows you the way to handle Virtuozzo OS and application templates properly – create and install templates and template updates on the Hardware Node, add them to and remove from Virtual Private Servers, cache OS templates, etc.

Chapter 5, *Managing EZ templates*, provides instructions on how to manage OS and application EZ templates having much in common with standard Virtuozzo OS and application templates and introduced in Virtuozzo 3.0 for the first time.

Chapter 6, *Managing Resources*, zeroes in on configuring and monitoring the resource control parameters for different VPSs. These parameters comprise disk quotas, network accounting and shaping, CPU and system resources. Common ways of optimizing your VPSs configurations are suggested at the end of the chapter.

Chapter 7, *Real-Time Monitoring in Virtuozzo*, explains the way to keep track of the consumption of all kind of resources by running Virtual Private Servers and the Hardware Node itself in real time. Configuring VZMC and the Monitor Node for sending alerts is part and parcel of the monitoring process.

Chapter 8, *Managing Services and Processes*, familiarizes you with the operations you can perform on processes and services in Virtuozzo by using both the command-line utilities and VZMC graphical interface.

Chapter 9, *Name-Based Hosting for Virtuozzo*, provides comprehensive information on the name-based hosting functionality of Virtuozzo 3.0.

Chapter 10, *Keeping Your Virtuozzo System Up-to-Date*, serves as a reference on the ways to keep all the software components of a Hardware Node up-to-date.

Chapter 11, *Customizing VZCC/VZPP*, contains instructions for experienced administrators on the way to customize VZCC and VZPP by changing their layout and appearance. Besides, it explains how to set up the demo mode to allow your potential customers to manage real Virtual Private Servers by means of VZCC/VZPP.

Chapter 12, *Advanced Tasks*, enumerates those tasks that are intended for advanced system administrators who would like to obtain deeper knowledge about Virtuozzo capabilities.

Chapter 13, *Mastering VZMC*, focuses on those tasks that are most comfortably accomplished using not the command-line utilities, but VZMC graphical interface.

Chapter 14, *Troubleshooting*, suggests ways to resolve common inconveniences should they occur during your work with the Virtuozzo software.

Documentation Conventions

Before you start using this guide, it is important to understand the documentation conventions used in it. For information on specialized terms used in the documentation, see the Glossary at the end of this document.

Typographical Conventions

The following kinds of formatting in the text identify special information.

Formatting convention	Type of Information	Example
Triangular Bullet(➤)	Step-by-step procedures. You can follow the instructions below to complete a specific task.	<i>To create a VPS:</i>
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Go to the QoS tab.
	Titles of chapters, sections, and subsections.	Read the Basic Administration chapter.
<i>Italics</i>	Used to emphasize the importance of a point, to introduce a term or to designate a command line placeholder, which is to be replaced with a real name or value.	These are the so-called <i>EZ templates</i> . To destroy a VPS, type <code>vzctl destroy <i>vpsid</i></code> .
Monospace	The names of commands, files, and directories.	Use <code>vzctl start</code> to start a VPS.
Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	Saved parameters for VPS 101
Monospace Bold	What you type, contrasted with on-screen computer output.	# rpm -V virtuozzo-release
CAPITALS	Names of keys on the keyboard.	SHIFT, CTRL, ALT
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another.	CTRL+P, ALT+F4

Shell Prompts in Command Examples

Command line examples throughout this guide presume that you are using the Bourne-again shell (bash). Whenever a command can be run as a regular user, we will display it with a dollar sign prompt. When a command is meant to be run as root, we will display it with a hash mark prompt:

Bourne-again shell prompt \$

Bourne-again shell root prompt #

General Conventions

Be aware of the following conventions used in this book.

- Chapters in this guide are divided into sections, which, in turn, are subdivided into subsections. For example, **Documentation Conventions** is a section, and **General Conventions** is a subsection.
- When following steps or using examples, be sure to type double-quotes ("), left single-quotes ('), and right single-quotes (') exactly as shown.
- The key referred to as RETURN is labeled ENTER on some keyboards.

The root path usually includes the `/bin`, `/sbin`, `/usr/bin` and `/usr/sbin` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute paths in the examples.

Feedback

If you spot a typo in this guide, or if you have thought of a way to make this guide better, we would love to hear from you!

If you have a suggestion for improving the documentation (or any other relevant comments), try to be as specific as possible when formulating it. If you have found an error, please include the chapter/section/subsection name and some of the surrounding text so we can find it easily.

Please submit a report by e-mail to userdocs@swsoft.com.

CHAPTER 2

Virtuozzo Philosophy

In This Chapter

About Virtuozzo Software	15
Distinctive Features of Virtuozzo	16
Main Principles of Virtuozzo Operation	20
Hardware Node Availability Considerations	30

About Virtuozzo Software

What is Virtuozzo

Virtuozzo is a complete server automation and virtualization solution based on a patent-pending technology developed by SWsoft. Virtuozzo creates multiple isolated Virtual Private Servers (VPSs) on a single physical server to share hardware, licenses, and management effort with maximum efficiency. Each VPS performs and executes exactly like a stand-alone server for its users and applications as it can be rebooted independently and has its own root access, users, IP addresses, memory, processes, files, applications, system libraries, and configuration files. Light overhead and efficient design of Virtuozzo makes it the right virtualization choice for production servers with live applications and real-life data.

The basic Virtuozzo capabilities are:

- **Dynamic Real-time Partitioning** – Partition a physical server into tens of VPSs, each with full dedicated server functionality.
- **Resource Management** – Assign and control VPS resource parameters and re-allocate resources in real-time.
- **Virtualization** - Move VPSs and its application(s) between physical servers transparently with near-zero or zero downtime.
- **Mass Management** - Manage a multitude of physical servers and Virtual Private Servers in a unified way.

Virtuozzo Applications

Virtuozzo is often bundled with HSPComplete, a comprehensive solution for Hosting Service Providers, based on the Virtuozzo technology. Virtuozzo allows Hosting Service Providers to:

- Have hundreds of customers with their individual full-featured virtual private servers (Virtual Private Servers) sharing a single physical server;
- Provide each customer with a guaranteed Quality of Service;
- Transparently move customers and their environments between servers, without any manual reconfiguration.

While Virtuozzo is effectively coupled with HSPComplete as well as with other hosting automation solutions, the scope of its application is not limited to them.

If you administer a number of Linux dedicated servers within an enterprise, each of which runs a specific service, you can use Virtuozzo to consolidate all these servers onto a single computer without losing a bit of valuable information and without compromising performance. Virtual Private Servers behave just like an isolated stand-alone server:

- Each VPS has its own processes, users, files and provides full root shell access;
- Each VPS has its own IP addresses, port numbers, filtering and routing rules;
- Each VPS can have its own configuration for the system and application software, as well as its own versions of system libraries. It is possible to install or customize software packages inside a VPS independently from other VPSs or the host system. Multiple distributions of a package can be run on one and the same Linux box.

In fact, hundreds of servers may be grouped together in this way. Besides the evident advantages of such consolidation (increased facility of administration and the like), there are some you might not even have thought of, say, cutting down electricity bills by times!

Virtuozzo proves invaluable for IT educational institutions that can now provide every student with a personal Linux server, which can be monitored and managed remotely. Software development companies may use virtual environments for testing purposes and the like.

Thus, Virtuozzo can be efficiently applied in a wide range of areas: web hosting, enterprise server consolidation, software development and testing, user training, and so on.

Distinctive Features of Virtuozzo

The concept of Virtuozzo Virtual Private Servers is distinct from the concept of traditional virtual machines in the respect that Virtual Private Servers (VPSs) always run the same OS kernel as the host system (Linux on Linux, Windows on Windows, etc.). This single-kernel implementation technology allows to run Virtual Private Servers with a near-zero overhead. Thus, Virtuozzo VPSs offer an order of magnitude higher efficiency and manageability than traditional virtualization technologies.

OS Virtualization

From the point of view of applications and Virtual Private Server users, each VPS is an independent system. This independency is provided by a virtualization layer in the kernel of the host OS. Note that only an negligible part of the CPU resources is spent on virtualization (around 1-2%). The main features of the virtualization layer implemented in Virtuozzo are the following:

- VPS looks like a normal Linux system. It has standard startup scripts, software from vendors can run inside VPS without Virtuozzo-specific modifications or adjustment;
- A user can change any configuration file and install additional software;
- Virtual Private Servers are fully isolated from each other (file system, processes, Inter Process Communication (IPC), `sysctl` variables);
- Virtual Private Servers share dynamic libraries, which greatly saves memory;
- Processes belonging to a VPS are scheduled for execution on all available CPUs. Consequently, VPSs are not bound to only one CPU and can use all available CPU power.

Network Virtualization

The Virtuozzo network virtualization layer is designed to isolate VPSs from each other and from the physical network:

- Each VPS has its own IP address; multiple IP addresses per VPS are allowed;
- Network traffic of a VPS is isolated from the other VPSs. In other words, Virtual Private Servers are protected from each other in the way that makes traffic snooping impossible;
- Firewalling may be used inside a VPS (the user can create rules limiting access to some services using the canonical `iptables` tool inside the VPS). In other words, it is possible to set up firewall rules from inside a VPS;
- Routing table manipulations are allowed to benefit from advanced routing features. For example, setting different maximum transmission units (MTUs) for different destinations, specifying different source addresses for different destinations, and so on.

Virtuozzo File System (VZFS)

VZFS is a file system that allows to share common files among multiple VPSs without sacrificing flexibility. It is possible for VPS users to modify, update, replace, and delete shared files. When a user modifies a shared file, VZFS creates a private copy of the file transparently for the user. Thus, the modifications do not affect the other users of the file. Main benefits of VZFS are the following:

- It saves memory required for executables and libraries. A typical VPS running a simple web site might consume around 20–30 MBytes of RAM just for executable images. Sharing this memory improves scalability and total system performance;
- It saves disk space. A typical Linux server installation occupies several hundred MBytes of disk space. Sharing the files allows you to save up to 90% of disk space;
- VZFS does not require having different physical partitions for different VPSs or creating a special “file system in a file” setup for a VPS. This significantly simplifies disk administration;
- Disk quota enables the administrator to limit disk resources available to a VPS on-the-fly, in the same manner as the standard disk quota system works on a per-user basis. Disk quota for users and groups inside VPSs is also supported.

Templates

A template (or a package set) in Virtuozzo is a set of original application files repackaged for mounting over Virtuozzo File System. Usually it is just a set of RPM packages for Red Hat like systems. Virtuozzo provides tools for creating templates, installing, upgrading, adding them to and removing them from a VPS. Using templates lets you:

- Share the RAM among similar applications running in different VPSs to save hundreds of megabytes of memory;
- Share the files comprising a template among different VPSs to save gigabytes of disk space;
- Deploy applications simultaneously in many VPSs;
- Use different versions of an application on different VPSs (for example, perform an upgrade only in some VPSs).

There are two types of templates in Virtuozzo. These are OS templates and application templates. An OS template is an operating system and the standard set of applications to be found right after the installation. Virtuozzo uses OS templates to create new Virtual Private Servers with a preinstalled operating system. An application template is a set of repackaged RPMs optionally accompanied with configuration scripts. Virtuozzo uses application templates to add extra software to the existing Virtual Private Servers. For example, you can create a Virtual Private Server on the basis of the `redhat9` OS template and add the MySQL application to it with the help of the `mysql` template.

For detailed information on Virtuozzo templates, see the [Understanding Templates](#) section.

Resource Management

Virtuozzo Resource Management controls the amount of resources available to Virtual Private Servers. The controlled resources include such parameters as CPU power, disk space, a set of memory-related parameters. Resource management allows Virtuozzo to:

- Effectively share available Hardware Node resources among VPSs;
- Guarantee Quality-of-Service (QoS) in accordance with a service level agreement (SLA);
- Provide performance and resource isolation and protect from denial-of-service attacks;
- Simultaneously assign and control resources for a number of Virtual Private Servers;
- Manage a multitude of Hardware Nodes in a unified way by means of Virtuozzo Management Console (VZMC);
- Collect usage information for system health monitoring, etc.

Resource Management is much more important for Virtuozzo than for a standalone computer since computer resource utilization in a Virtuozzo-based system is considerably higher than that in a typical system.

Main Principles of Virtuozzo Operation

Basics of Virtuozzo Technology

In this section we will try to let you form a more or less precise idea of the way the Virtuozzo software operates on your computer. Please see the figure below:

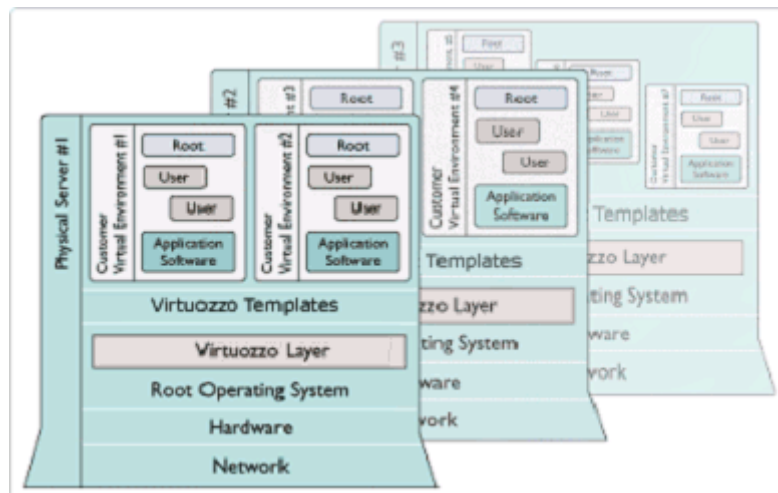


Figure 1: Virtuozzo Technology

This figure presumes that you have a number of physical servers united into a network. In fact, you may have only one dedicated server to effectively use Virtuozzo for the needs of your network. If you have more than one Virtuozzo-based physical server, each one of the servers will have a similar architecture. In Virtuozzo terminology, such servers are called *Hardware Nodes* (or *HN*, or just *Nodes*), because they represent hardware units within a network.

Virtuozzo 3.0 is installed on Red Hat Enterprise Linux 4, Fedora Core 2 or 4, CentOS 4, and Suse Linux Enterprise Server 9 configured in a certain way. For example, such customized configuration shall include the creation of a `/vz` partition, which is the basic partition for hosting Virtual Private Servers and which must be way larger than the root partition. This and similar configuration issues are most easily resolved during Linux installation on the Hardware Node. Detailed instructions on installing Linux (called *Host Operating System*, or *Root Operating System* in Figure 1) on the Hardware Node are provided in the next chapter.

Virtuozzo is installed in such a way that you will be able to boot your computer either with Virtuozzo support or without it. This support is presented as “linux virtuozzo” in your boot loader and shown as *Virtuozzo Layer* in the figure above.

However, at this point you are not yet able to create Virtual Private Servers. A *Virtual Private Server* is functionally identical to an isolated standalone server, having its own IP addresses, processes, files, users, its own configuration files, its own applications, system libraries, and so on. Virtual private servers share the same *Hardware Node* and the same OS kernel. However, they are isolated from each other. A Virtual Private Server is a kind of ‘sandbox’ for processes and users.

Different Virtual Private Servers can run different versions of Linux (for example, SuSE 9.3 or Fedora Core 4 and many others). Each VPS can run its own version of Linux. In this case we say that a VPS is based on a certain OS template. OS templates are RPM packages shipped with Virtuozzo 3.0. Before you are able to create a Virtual Private Server, you should install the corresponding OS template in Virtuozzo. This is displayed as *Virtuozzo Templates* in the scheme above.

After you have installed at least one OS template, you can create any number of VPSs with the help of standard Virtuozzo utilities, configure their network and/or other settings, and work with these VPSs as with fully functional Linux servers.

Understanding Templates

Templates are part and parcel of the Virtuozzo philosophy because they provide a way of sharing resources among lots of VPSs, thus enabling huge savings in terms of disk space and memory. For example, when you install an OS template in Virtuozzo, Virtuozzo creates a `/vz/template/<name_of_the_OS>` directory containing all the OS files that can be shared among VPSs. When a VPS based on this template is created, it contains only symlinks to the OS template files. These symlinks occupy very little space on the hard disk. They are situated in the so-called *private area* of the VPS. The corresponding directory is `/vz/private/<VPS_ID>`. The private area of a VPS contains not only symlinks to the necessary template files, but also the copy-on-write area of the VPS (the area for storing the information about those changes that the VPS makes to the template files; this information pertains only to the given directory) and all the private VPS files. When the VPS is started, this private area is mounted as Virtuozzo File System (VZFS) to the `/vz/root/<VPS_ID>` directory. This directory is seen as the root directory from within the VPS. And, which is the pivot of it all, thanks to the VZFS, the symlinks of the VPS private area are seen as real files there!

Thus, the most important directories in the `/vz` partition are the following:

- | | |
|---------------------------|--|
| <code>/vz/template</code> | - contains OS and application files shared among VPSs; |
| <code>/vz/private</code> | - contains VZFS symlinks to template files; |
| <code>/vz/root</code> | - contains VPS mounted symlinks and all other VPS files. |

The relation of these directories may be represented as below:

`/vz/template` (real files) → `/vz/private` (symlinks) → `/vz/root` (symlinks seen as real files in `/` for the VPS)

While you are able to perform all kinds of tasks within a Virtual Private Server including building RPM packages and installing them, Virtuozzo provides an easy and far more efficient way of installing the applications you need on VPSs. In the same way as you install an OS template on the Virtuozzo system in order to create any number of Virtual Private Servers on its basis and share its resources, you can install application templates in Virtuozzo in order to share application files among any number of VPSs. You can then add these applications to any number of Virtual Private Servers with a single command.

It goes without saying that in case you want to install an application on only one VPS, there is no need in working with templates: you can as well work inside the corresponding VPS.

Thus, in order to install an application on two or more VPSs, you should perform the following steps:

- 1** Create a Virtuozzo template. This means repackaging a set of original vendor-distributed RPM files. This repackaging is simply done with the help of a standard Virtuozzo utility, `vzpkgcreat`.
- 2** Install the created Virtuozzo template on the Hardware Node. Upon installation, the corresponding `/vz/template/<application_name>` directory is created on the Hardware Node in order to enable sharing common application files among VPSs. This is done with the standard Linux `rpm -i` command.
- 3** Add the installed template to any number of Virtual Private Servers of your choice with the help of the `vzpkgadd` command.

All these steps can also be performed not by means of Linux command-line utilities but with the help of Virtuozzo Management Console (VZMC) - a graphical means for managing Virtuozzo Hardware Nodes. An overview of VZMC will be given later in this section.

Virtuozzo also provides you with the ability to create and install *template updates* for the templates already installed on the HN and/or added to VPSs. Naturally, it is also very easy to remove templates or only their updates from VPSs and/or from the Hardware Node.

Now imagine the situation a VPS user modifies a file that comprises a template and thus is used by more than one VPS. In this case Virtuozzo, naturally, does not modify the template file. It uses a number of means to record the modifications of this file for the given VPS, including using a special `/vz/private/<VPS_ID>/cow` directory. With these modifications thus recorded, the corresponding symlink from the VPS private area is mounted with its modifications and is seen as a real file from within the VPS, as before. In other cases, some symlink may be simply replaced with the modified file.

Differences Between OS and Application Templates

Actually, there are four major differences between OS templates and application templates:

- OS templates are used to create new Virtual Private Servers, whereas application templates provide additional software for already created VPSs.
- OS templates may and usually do use action scripts, whereas application templates cannot use action scripts in the current version of Virtuozzo.
- You may define a list of compatible templates and a list of required templates for application templates.
- OS templates and their updates are cacheable, whereas application templates and their updates are not.

The last point needs further explanation. The fact is that creating a huge number of symlinks to the Operating System template when creating a VPS (i.e. its private area) may take a very considerable amount of time. To reduce the time needed for creating a new VPS, the `vzpkgcache` utility allows you to create a tarball of the potential private area of a VPS based on the corresponding template. This tarball is also located in the `/vz/template` directory. When a VPS is being created, the tarball is simply deployed into the VPS private area.

Virtuozzo Configuration

Virtuozzo 3.0 allows you to flexibly configure various settings for the Virtuozzo system in general as well as for each and every Virtual Private Server. Among these settings are disk and user quota, network parameters, default file locations and configuration sample files, and others.

Virtuozzo stores the configuration information in two types of files: the global configuration file `/etc/sysconfig/vz` and VPS configuration files `/etc/sysconfig/vz-scripts/<VPS_ID>.conf`. The global configuration file defines global and default parameters for VPS operation, for example, logging settings, enabling and disabling disk quota for VPSs, the default configuration file and OS template on the basis of which a new VPS is created, and so on. On the other hand, a VPS configuration file defines the parameters for a given particular VPS, such as disk quota and allocated resources limits, IP address and host name, and so on. In case a parameter is configured both in the global Virtuozzo configuration file, and in the VPS configuration file, the VPS configuration file takes precedence. For a list of parameters constituting the global configuration file and the VPS configuration files, turn to the [Virtuozzo Reference Guide](#).

The configuration files are read when Virtuozzo and/or VPSs are started. However, Virtuozzo standard utilities, for example, `vzctl`, allow you to change many configuration settings “on-the-fly”, either without modifying the corresponding configuration files or with their modification (if you want the changes to apply the next time Virtuozzo and/or VPSs are started).

Some Virtuozzo utilities have their own configuration files. For example, `vzbackup`, which is responsible for backing up VPS private areas and configuration files, has its own global configuration file `/etc/vzbackup.conf` and may have a number of per-Node configuration files located in the backup directory. This directory is defined in the backup global configuration file. Both the global backup configuration file and per-Node ones are located on a central “backup” node. There are a number of other specific configuration files. All of them are detailed in the [Configuring Virtuozzo](#) chapter of the [Virtuozzo Reference Guide](#).

Understanding Licensing

Every computer running VPSs shall have its own license file. License files (or simply “licenses”) are issued by SWsoft and define the number of VPSs the computer (Hardware Node) may run and the maximal CPU power that the Node may use (i.e. computers with more powerful CPUs than is stated in the license will not be allowed to run Virtuozzo). Licenses can be time-limited or permanent. It is prohibited to share a license with another customer.

Virtuozzo licenses have a start date and, if they are time-limited, may also have an expiration date specified in them. You shall have to set up your system clock correctly; otherwise, the license validation may fail.

The Virtuozzo license can be obtained from SWsoft by sending a license request to your SWsoft sales contact (e.g. sales@swsoft.com). A license request is a file generated by the `vzlicreq` utility. This utility creates a file named `vzLicRequest` in the directory you run the utility in. This file should be sent to your SWsoft representative.

Thereafter, SWsoft provides you with the corresponding license file(s) named `vzlicense` and you can copy it (them) to the `/etc/hspc/licenses` directory on the corresponding Hardware Node(s). After you have copied the file(s) you received, you must run the `vzlicload` utility to load the license into the Virtuozzo kernel.

VZMC Overview

Virtuozzo Management Console (VZMC) is a remote management tool for Virtuozzo with graphical user interface. VZMC is designed for Hardware Node administrators having access to all the VPSs on a particular Node. It allows the administrator to control multiple Hardware Nodes, to manage all sorts of Virtual Private Servers, and to monitor the system.

VZMC Specific Features

VZMC provides tools for managing any number of Hardware Nodes and Host Operating Systems, including the following:

- Clusters of Hardware Nodes with unified space of VPS IDs and IP addresses;
- Global Virtuozzo configuration parameters;
- Services of the Host OS;
- Users and groups;
- Disk usage;
- Network bandwidth usage;
- Network traffic accounting;
- Mount points;
- Firewall configuration.

VZMC facilitates major operations on all kinds of Virtual Private Servers such as their:

- Creating and recovering;
- Starting, stopping, and deleting;
- Backing up and restoring ;
- Migrating.

VZMC also provides flexible means for managing various VPS parameters, among which there are:

- Files;
- Services;
- Users and groups;
- Network settings ;
- Action scripts;
- Mount points;
- Firewall configuration.

VZMC may monitor Virtual Private Servers as well as Hardware Nodes. It also provides access to various system logs. Alerts notify you of lack of resources or system failures.

- VZMC supports all the Virtuozzo template operations, facilitating:
- Creating templates and/or template updates;
- Uploading and installing templates and/or template updates on the Hardware Node;
- Adding/removing templates and/or template updates to/from Virtual Private Servers.

VZMC Network Architecture

VZMC uses a typical client/server architecture. The client VZMC program runs on either Microsoft Windows 2000/XP/2003 or Linux (Red Hat 9; Fedora Core 1, 2, 3, and 4; Red Hat Enterprise Linux 3.0 and 4.0; CentOS 3 and 4) workstation with X Window System.

The client application with the graphical user interface connects to the `vzagent` software, which is running in the special Service VPS on the Hardware Node. `vzagent` communicates with the client via the well-documented open VZAgent XML API and controls the Hardware Node itself and Virtual Private Servers.

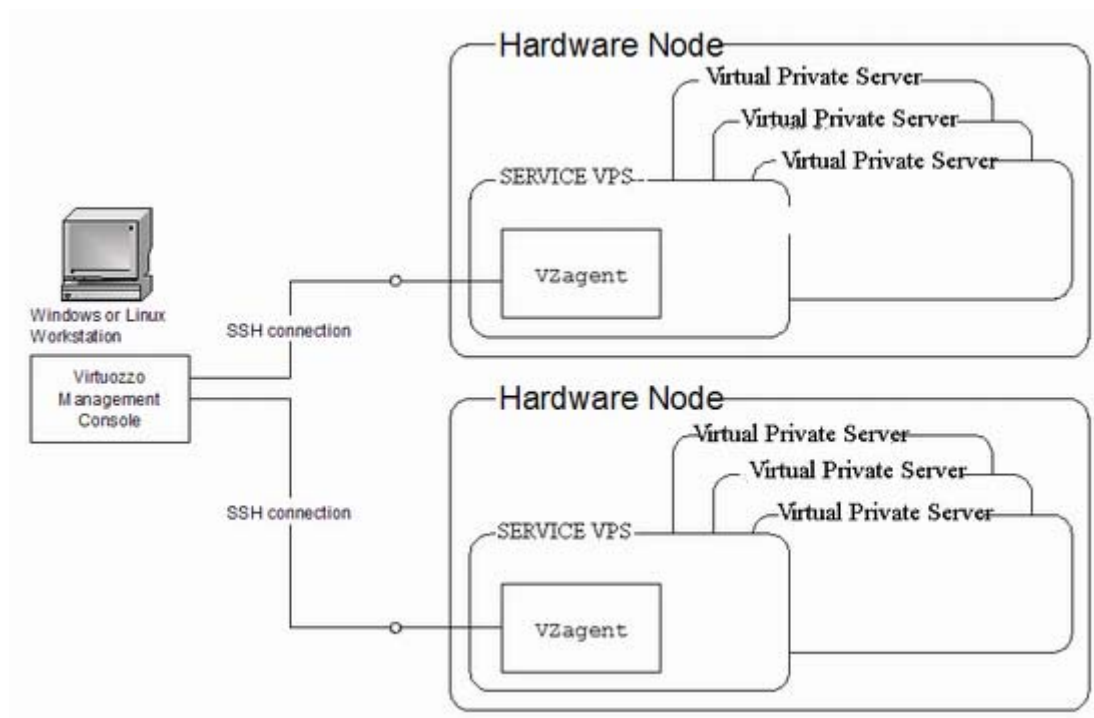


Figure 2: VZMC Network Architecture

The client may control multiple Hardware Nodes simultaneously by connecting to multiple agents as is shown in the figure above. As the communications between the client and `vzagents` are secure, the VZMC workstation may be located virtually anywhere on the net.

Hardware Node Main Window

You will feel most comfortable with VZMC with the screen resolution of 1024x768 or higher. The main window of VZMC consists of two parts: the tree pane on the left, and view pane on the right. There is a list of Hardware nodes in the tree pane. The Hardware Node subtree represents various aspects of its management, e.g. Logs, Services, Templates, etc. The content of the view pane depends on the selected item in the tree pane.

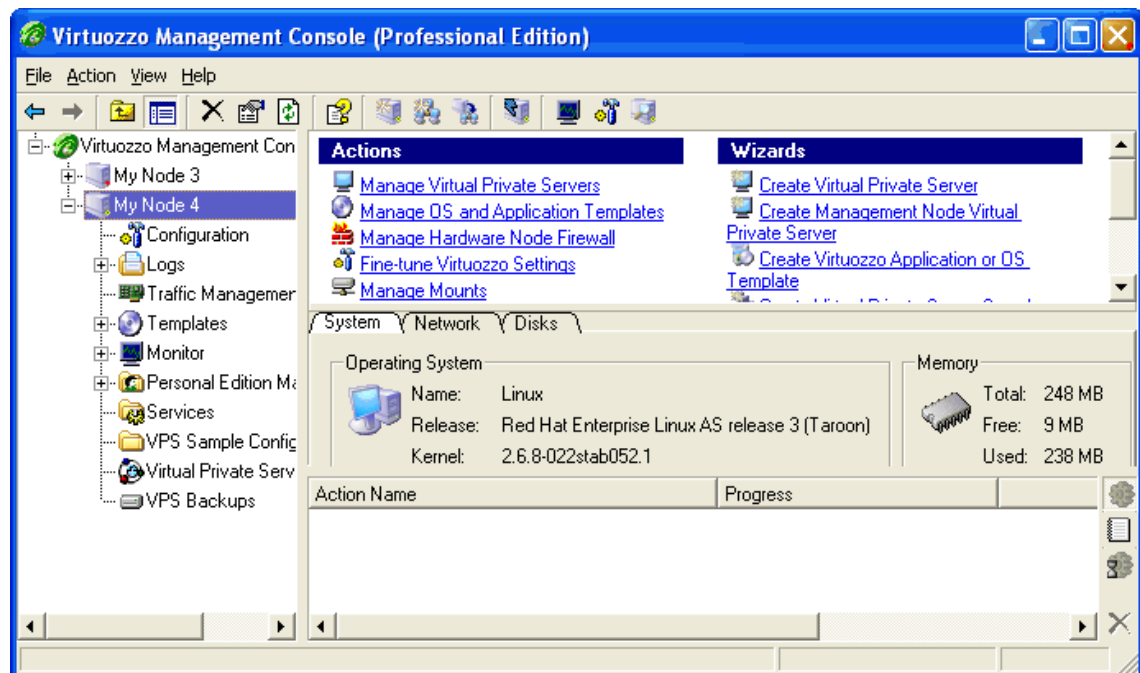


Figure 3: VZMC Main Window

Below the view pane on the right, there is also a small Actions/Messages/Operations pane. You may switch between the Actions and Messages modes by clicking buttons to the right of this pane. The Actions pane displays the progress of VZMC actions. The Messages pane displays the detailed diagnostics of various VZMC errors. The Operations pane shows the result of various asynchronous tasks performed with Virtual Private Servers.

You can view the summary page for every Hardware Node. Click on the name of the Hardware Node you are interested in in the tree in the left pane of the VZMC main window or double-click the name of the Hardware Node in the list of Nodes in the right pane.

The upper part of the view pane contains shortcuts to the most important tasks you are likely to do. However, all the actions and operations are accessible via the VZMC toolbar, Action menu, and context menus. The bottom part of the view pane includes three tabs: System, Network, and Disks. The System tab describes the OS distribution and kernel version, CPU(s), RAM, and swap information. The Network tab describes the Hardware Node network configuration: interfaces, DNSs, and IP addresses. The Disks tab describes disks available on the Hardware Node and their utilization.

Note: On the picture above, the summary page for a Hardware Node running the Linux operating system is shown. The summary page for Hardware Nodes running Windows 2003 Server may slightly differ in respect of the OS installed on them, the number of operations that can be performed on the Node, etc.

VZCC Overview

Virtuozzo Control Center (VZCC) is designed for Hardware Node administrators and provides you with the ability to manage a particular Hardware Node and all Virtual Private Servers residing on it with the help of a standard Web browser on any platform. A list of supported browsers is given below:

- Internet Explorer 6.0 and above;
- Mozilla 1.7 and above;
- Firefox 1.0 and above.

Chances are that you will also be able to use other browsers, but Virtuozzo has not been extensively tested with them.

In the current version of Virtuozzo, you are able to perform the following main operations on the Hardware Node and in the VPS context by using VZCC:

- View and install Virtuozzo licenses;
- Create new Virtual Private Servers;
- Manage VPS backups;
- Edit the VPS hostname, IP address, and DNS server parameters;
- Manage VPS applications;
- Manage VPS configuration templates;
- Manage both VPS groups and individual VPSs;
- Manage and monitor Hardware Node and VPS resources;
- Start, stop, and restart certain services inside the VPS;
- View the processes currently running on the Node and inside Virtual Private Servers and send signals to them;
- View the Virtuozzo logs;
- Customize the VZCC interface, etc.

Note: Detailed information on Virtuozzo Control Center is given in its comprehensive online help system and the **VZCC Administrator's Guide** delivered with VZCC.

VZPP Overview

Wherever Virtuozzo is applied, there are people that are supposed to be administrators of particular VPSs only, with no access rights to Hardware Nodes as such. This is only but natural as it corresponds directly with the concept of a virtualization technology. Such people can be subscribers to a hosting provider, university students, or administrators of a particular server within an enterprise. Virtuozzo 3.0 is equipped with a web-based tool for managing personal VPSs called Virtuozzo Power Panels (VZPP).

Virtuozzo Power Panels is a means for administering personal VPSs thru a common browser - Internet Explorer, Mozilla, and others. It is implemented by the `vzcp` package installed inside the Service VPS during the Virtuozzo installation. The `vzcpcon` process running in the Service VPS handles the client browser requests and passes them to the `vzagent` software, which is responsible for managing all the VPSs of the given Hardware Node.

VZPP allows VPS administrators to:

- Start, stop, or restart the VPS;
- Repair the VPS;
- Reinstall the VPS;
- Back up and restore the VPS;
- Change the VPS root password;
- Start, stop, or restart certain services inside the VPS;
- Access other control panels installed in the VPS, for example the Plesk control panel.
- View the list of VPS processes and send them signals;
- View the current resources consumption and resources overusage alerts;
- View the Virtuozzo logs, etc.

Access rights to administer particular VPSs by means of VZPP are determined by the Hardware Node administrator. Detailed instructions on how to control access rights to particular VPSs thru VZPP are provided in the **Setting VZCC/VZPP to Work** chapter of the **Virtuozzo Installation Guide**.

Note: VZPP can also be used by the Hardware Node administrator for managing any Virtual Private Server on the given Node.

Hardware Node Availability Considerations

Hardware Node availability is more critical than the availability of a typical PC server. Since it runs multiple Virtual Private Servers providing a number of critical services, Hardware Node outage might be very costly. Hardware Node outage can be as disastrous as the simultaneous outage of a number of servers running critical services.

In order to increase Hardware Node availability, we suggest you follow the recommendations below:

- Use RAID storage for critical VPS private areas. Do prefer hardware RAID, but software mirroring RAID might suit too as a last resort.
- Do not run software on the Hardware Node itself. Create special Virtual Private Servers where you can host necessary services such as BIND, FTPD, HTTPD, and so on. On the Hardware Node itself, you need only the SSH daemon. Preferably, it should accept connections from a pre-defined set of IP addresses only.
- Do not create users on the Hardware Node itself. You can create as many users as you need in any Virtual Private Server. Remember, compromising the Hardware Node means compromising all Virtual Private Servers as well.

CHAPTER 3

Operations on Virtual Private Servers

This chapter describes how to perform day-to-day operations on separate Virtual Private Servers taken in their wholeness.

Note: We assume that you have successfully installed, configured, and deployed your Virtuozzo system. In case you have not, please turn to the **Virtuozzo Installation Guide** providing detailed information on all these operations.

In This Chapter

Creating New Virtual Private Server	31
Configuring Virtual Private Server	44
Starting, Stopping, Restarting, and Querying Status of Virtual Private Server	47
Listing Virtual Private Servers	49
Migrating Virtual Private Server	51
Backing Up and Restoring Virtual Private Servers	56
Reinstalling Virtual Private Server	67
Deleting Virtual Private Server	71
Disabling VPS	73
Running Commands in Virtual Private Server	74

Creating New Virtual Private Server

This section guides you through the process of creating a Virtual Private Server. Virtuozzo allows you to create VPSs in one of the following ways:

- By using the technology of Virtuozzo OS templates. Detailed information on OS templates is provided in the **Understanding Templates** subsection (on page 21). In this case we assume that you have successfully installed Virtuozzo and at least one OS template. If there are no OS templates installed on the Hardware Node, turn to the **Managing Templates** chapter first.
- By using the technology of Virtuozzo OS EZ templates. This functionality is available beginning with Virtuozzo 3.0. For more information on OS EZ templates, please see the **Understanding EZ Templates** subsection (on page 93). In this case we assume that you have successfully installed Virtuozzo and prepared at least one OS EZ templates. If there are no OS EZ templates prepared for the VPS creation, turn to the **Managing EZ Templates** chapter first.

Before You Begin

Before you start creating a Virtual Private Server, you should:

- Check that the Hardware Node is visible on your network. You should be able to connect to/from other hosts. Otherwise, your Virtual Private Servers will not be accessible from other computers.
- Check that you have at least one IP address per Virtual Private Server and the addresses belong to the same network as the Hardware Node or routing to the Virtual Private Servers has been set up via the Hardware Node.

To create a new Virtual Private Server, you have to:

- choose the new Virtual Private Server ID;
- choose the OS template to use for the Virtual Private Server;
- create the Virtual Private Server itself.

Choosing Virtual Private Server ID

Every Virtual Private Server has a numeric ID, also known as VPS ID, associated with it. The ID is a 32-bit integer number beginning with zero and unique for a given Hardware Node. When choosing an ID for your Virtual Private Server, please follow the simple guidelines below:

- ID 0 is used for the Hardware Node itself. You cannot and should not try to create a Virtual Private Server with ID 0.
- This version of Virtuozzo uses ID 1 for the Service VPS.
- Virtuozzo reserves the IDs ranging from 0 to 100. Though Virtuozzo uses only IDs 0 and 1 from them, the next version might use additional Virtual Private Servers IDs for internal needs. *To facilitate upgrading, please do not create Virtual Private Servers with IDs below 101.*

The only strict requirement for a VPS ID is to be unique for a particular Hardware Node. However, if you are going to have several computers running Virtuozzo, we recommend assigning different VPS ID ranges to them. For example, on Hardware Node 1 you create Virtual Private Servers within the range of IDs from 101 to 1000; on Hardware Node 2 you use the range from 1001 to 2000, and so on. This approach makes it easier to remember on which Hardware Node a Virtual Private Server has been created, and eliminates the possibility of VPS ID conflicts when a Virtual Private Server migrates from one Hardware Node to another.

Another approach to assigning VPS IDs is to follow some pattern of VPS IP addresses. Thus, for example, if you have a subnet with the 10.0.x.x address range, you may want to assign the 17015 ID to the VPS with the 10.0.17.15 IP address, the 39108 ID to the VPS with the 10.0.39.108 IP address, and so on. This makes it much easier to run a number of Virtuozzo utilities eliminating the necessity to check up the VPS IP address by its ID and similar tasks. You can also think of your own patterns for assigning VPS IDs depending on the configuration of your network and your specific needs.

Before you decide on a new VPS ID, you may want to make sure that no VPS with this ID has yet been created on the Hardware Node. The easiest way to check whether the VPS with the given ID exists is to issue the following command:

```
# vzlist -a 101
VPS not found
```

This output shows that Virtual Private Server 101 does not exist on the particular Hardware Node; otherwise it would be present in the list.

If you use VZMC, click on the name of your Hardware Node in the left pane, and then on the Virtual Private Servers item. The VZMC right pane will display a list of existing VPSs on the Node.

WARNING! When deciding on a VPS ID, do not use the ID of any VPS that was ever present in the system unless you are sure that no data belonging to the old VPS remains on the Node. The fact is that the administrator of the newly-created VPS might have access to these data in this case, i.e. to the backups of the old VPS, its logs, statistics, etc.

Creating VPS on Basis of OS Standard Template

This subsection provides information on how you can create new Virtual Private Servers by using Virtuozzo OS standard templates. If you are running the 3.0 or newer version of Virtuozzo and going to use an OS EZ template to base your VPS on, you can skip this subsection and proceed to the [Creating VPS on Basis of OS EZ Template](#) subsection (on page 37).

Choosing OS Template

Before starting to create your Virtual Private Server, you shall decide on which OS template you want to base the new VPS. There might be several OS templates installed on the Hardware Node; use the `vzpkgls` command to find out the templates installed on your system:

```
# vzpkgls
redhat-9 20040116
redhat-as3-minimal 20040119
proftpd-rh9 20040119
```

As you can guess from the names, `redhat-9` and `redhat-as3-minimal` are OS templates, whereas `proftpd-rh9` is an application template. You can run an additional checkup with the `vzpkginfo` command to single out OS templates from application templates with confusing names; OS templates always have the `cached` section of their configuration file set to `yes`:

```
# vzpkginfo -g cached -b redhat-9
yes
# vzpkginfo -g cached -b proftpd-rh9
no
```

OS templates are pre-cached, so ‘yes’ in the output means that this is an OS template indeed. You can also list packages included into the template by running the `vzpkginfo -g base_version -b redhat-9` command (for listing the packages of the base version) or `vzpkginfo -p -b redhat-9` (for listing the packages of the current version). You can consult the [Virtuozzo Reference Guide](#) for complete information on the `vzpkginfo` command.

Note: The `redhat-as3-minimal` OS template is used to create the Service VPS only. You should use other OS templates installed on the Hardware Node to create regular Virtual Private Servers on their basis.

In VZMC, things are much simpler: just click consecutively the name of your Node, the **Templates** item, the **Standard Templates** item, and then the **OS Templates** item to see a list of the installed OS templates:

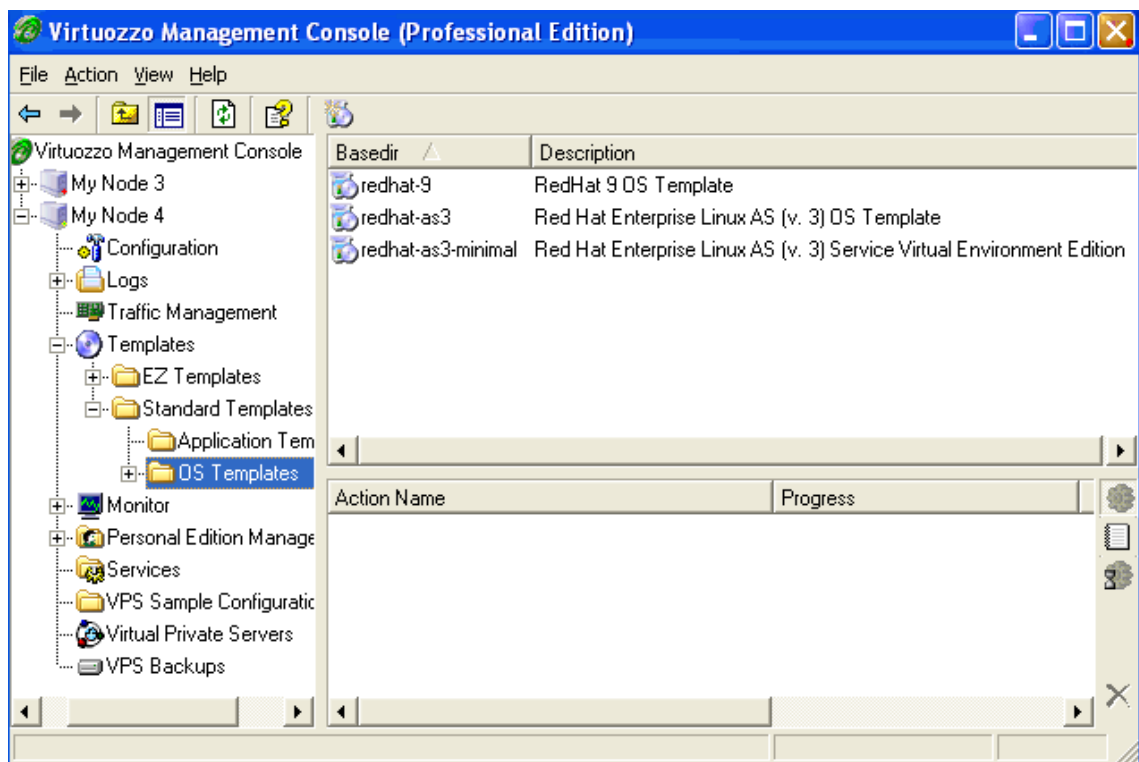


Figure 4: VZMC - Listing Templates

Note: If you have upgraded to Virtuozzo 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

Creating Virtual Private Server

After the VPS ID and the installed OS template have been chosen, you can create the VPS private area with the `vzctl create` command. The private area is the directory containing the VZFS symlinks, copy-on-write area, and private files of the given VPS. The private area is mounted to the `/vz/root/VPS_ID` directory on the Hardware Node and provides VPS users with a complete Linux file system tree.

The `vzctl create` command requires only the VPS ID and the name of the OS template as arguments; however, in order to avoid setting all the VPS resource control parameters after creating the private area, you can specify a sample configuration to be used for your new Virtual Private Server. The sample configuration files are residing in the `/etc/sysconfig/vz-scripts` directory and have names with the following mask: `vps-<configname>.conf-sample`. The most commonly used sample is the `ve-vps.basic.conf-sample` file; this sample file has resource control parameters suitable for most web site Virtual Private Servers.

Thus, for example, you can create a new VPS by typing the following string:

```
# vzctl create 101 --pkgset redhat-9 --config vps.basic
Creating VPS private area
VPS private area was created
```

In this case, Virtuozzo will create a Virtual Private Server with ID 101, the private area based on the `redhat-9` OS template, and configuration parameters taken from the `ve-vps.basic.conf-sample` sample configuration file.

If you specify neither an OS template nor a sample configuration, `vzctl` will try to take the corresponding values from the global Virtuozzo configuration file `/etc/sysconfig/vz`. So you can set the default values in this file using your favorite text file editor, for example:

```
DEF_OSTEMPLATE="fedora-core-4"
CONFIGFILE="vps.basic"
```

and do without specifying these parameters each time you create a new VPS.

Now you can create a VPS with ID 101 with the following command:

```
# vzctl create 101
Creating VPS private area
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
```

In principle, now you are ready to start your newly created Virtual Private Server. However, typically you need to set its network IP address, hostname, DNS server address and root password before starting the Virtual Private Server for the first time. Please see the next subsection for information on how to perform these tasks.

Creating VPS on Basis of OS EZ Template

The given subsection explains you the way to create new Virtual Private Servers by using OS EZ templates. This functionality is available beginning with Virtuozzo 3.0 only; so, if you are running a Virtuozzo version older than 3.0, you should use standard OS templates described in the previous subsection to base your VPSs on.

Choosing OS EZ Template

Before starting to create a Virtual Private Server, you shall decide on which OS EZ template your VPS will be based on. There might be several OS EZ templates installed on the Hardware Node and prepared for the VPS creation; use the `vzpkg list` command to find out what OS EZ templates are available on your system:

```
# vzpkg list -O
redhat-as4-x86          2005-11-21 23:59:44
fedora-core-4-x86      2005-12-20 12:45:52
```

The `-O` option passed to the `vzpkg list` command allows you to list only OS EZ templates installed on the Hardware Node. As you can see, the `redhat-as4-x86` and `fedora-core-4-x86` OS EZ templates are currently available on the Node. The time displayed beyond OS EZ templates indicates when the corresponding EZ template was cached.

You can also use the `--with-summary` option to display brief information on the installed OS EZ templates:

```
# vzpkg list -O --with-summary
redhat-as4-x86          Red Hat Enterprise Linux AS v. 4 OS template
fedora-core-4-x86      Fedora Core 4 OS template
```

For complete information on the `vzpkg list` command, you can consult the [Virtuozzo Reference Guide](#).

In VZMC, you only have to click consecutively the name of your Node, the **Templates** item, the **EZ Templates** item, and the **OS Templates** item to see a list of the installed OS EZ templates:

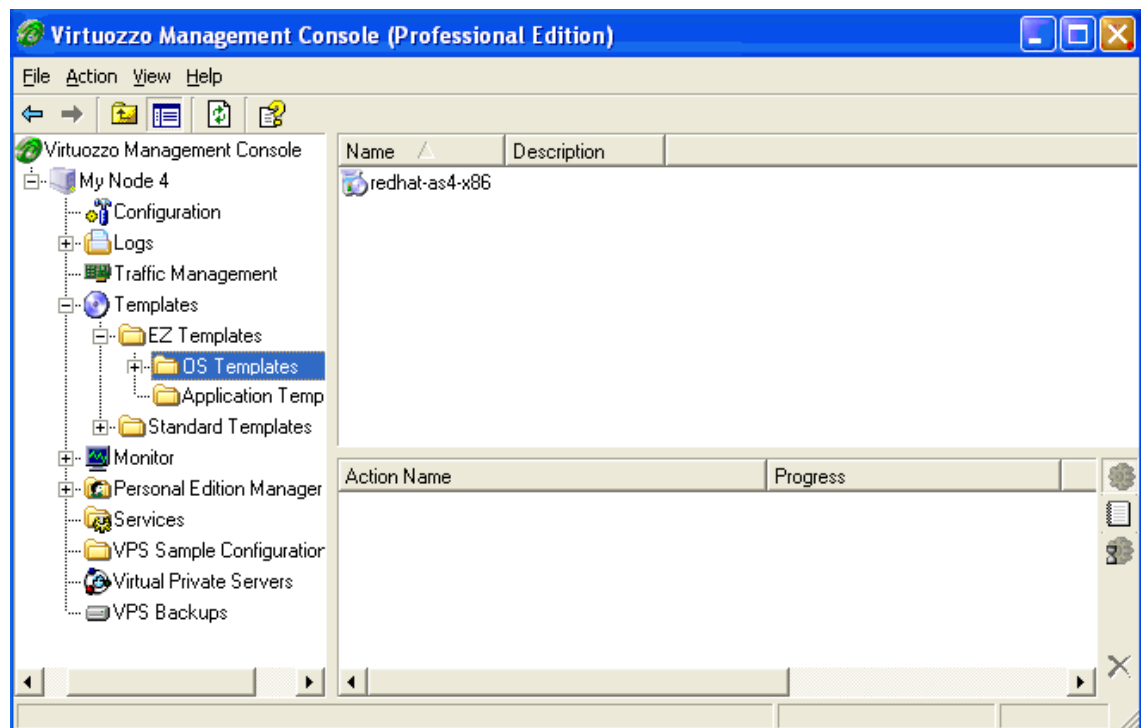


Figure 5: VZMC - Listing EZ OS templates

Note: If you have upgraded to Virtuozzo 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

Creating Virtual Private Server

After the VPS ID and the installed OS EZ template have been chosen, you can create the VPS private area with the `vzctl create` command. The private area is the directory containing the VZFS symlinks, copy-on-write area, and private files of the given VPS. The private area is mounted to the `/vz/root/VPS_ID` directory on the Hardware Node and provides VPS users with a complete Linux file system tree.

The `vzctl create` command requires only the VPS ID and the name of the OS template as arguments; however, in order to avoid setting all the VPS resource control parameters after creating the private area, you can specify a sample configuration to be used for your new Virtual Private Server. The sample configuration files are residing in the `/etc/sysconfig/vz-scripts` directory and have names with the following mask: `vps-<configname>.conf-sample`. The most commonly used sample is the `ve-vps.basic.conf-sample` file; this sample file has resource control parameters suitable for most Virtual Private Servers.

Thus, for example, you can create a new VPS by typing the following string:

```
# vzctl create 101 --ostemplate redhat-as4-x86 --config vps.basic
Creating VPS private area (redhat-as4-x86)
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
Delete port redirection
Adding port redirection to VPS(1): 4643 8443
```

In this case, Virtuozzo will create a Virtual Private Server with ID 101, the private area based on the `redhat-as4-x86` OS EZ template, and configuration parameters taken from the `ve-vps.basic.conf-sample` sample configuration file.

If you specify neither an OS template nor a sample configuration, `vzctl` will try to take the corresponding values from the global Virtuozzo configuration file (`/etc/sysconfig/vz`). So you can set the default values in this file using your favorite text file editor, for example:

```
DEF_OSTEMPLATE=".redhat-as4-x86"
CONFIGFILE="vps.basic"
```

and do without specifying these parameters each time you create a new VPS. Please keep in mind that the `.` symbol before the template name in the `DEF_OSTEMPLATE` parameter is used to indicate that the VPS being created is to be based on an EZ, and not standard template.

Now you can create a VPS with ID 101 with the following command:

```
# vzctl create 101
Creating VPS private area (redhat-as4-x86)
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
Delete port redirection
Adding port redirection to VPS(1): 4643 8443
```

In principle, now you are ready to start your newly created Virtual Private Server. However, typically you need to set its network IP address, hostname, DNS server address and root password before starting the Virtual Private Server for the first time.

Creating Virtual Private Servers in VZMC

VZMC uses one wizard both to create a Virtual Private Server and to initially configure it. You can launch this wizard by selecting the **Virtual Private Servers** item in the left pane and choosing the **Create New Virtual Private Server** option from the **Action** menu:

Create New Virtual Private Servers

Specify Basic Virtual Private Server Parameters

The wizard needs to know basic parameters for Virtual Private Server creation, such as sample configuration to use, number of VPSs to create, their IDs, etc

Please choose one of sample configuration as a basis to create your Virtual Private Servers:

Name	OS Template	Description
db.oracle		Configuration file for running Oracl...
vps.1024MB		Configuration file for allocating 10...
vps.2048MB		Configuration file for allocating 20...
vps.256MB		Configuration file for allocating 25...
vps.512MB		Configuration file for allocating 51...
vps.basic		Basic configuration file
vps.confxxx		Configuration file for running Confi...

Number of Virtual Private Servers to create:

Virtual Private Server ID

☐ Assign Virtual Private Server ID automatically from the pool at the Master node.

☒ Virtual Private Server ID start from:

Hostname

☒ Assign hostname automatically

☐ Hostname:

Root password

Password:

Confirm password:

Help < Back Next > Finish Cancel

Figure 6: VZMC - Creating New Virtual Private Server

By default, this is the only page of the wizard. The fact is all the VPS parameters including the templates, resource management parameters, ID and IP address can be retrieved on the basis of the VPS configuration sample indicated in the very first option and the cluster settings, if the given Hardware Node belongs to a cluster of Nodes. To know more about clusters, turn to the **Managing Node Clusters** section of the **Mastering VZMC** chapter. In particular, the **Assign Virtual Private Server ID automatically...** checkbox, if selected, tells VZMC to retrieve VPS ID(s) from the cluster database.

The **Hostname** group of options on the first page of the wizard shown above might help you make use of your DNS server. If your DNS server has records for the IP addresses that will be assigned to the newly-created VPSs, select the **Assign hostname automatically** radio button. The hostnames will be assigned on the basis of DNS records found.

By default, the root account is disabled in a newly-created VPS. To enable this account, you may enter the root password on the first page of the wizard. If you leave the **Password** and **Confirm password** fields blank, the root account will remain disabled.

If you do not rely on any configuration sample, press the **Next** button instead of **Finish**. In this case you will have to go through a number of steps of the wizard and set all the parameters of the new VPS(s) separately. If the Hardware Node where new Virtual Private Servers are being created belongs to a cluster, the next page of the wizard will be **Assign IP Addresses to Virtual Private Servers**:

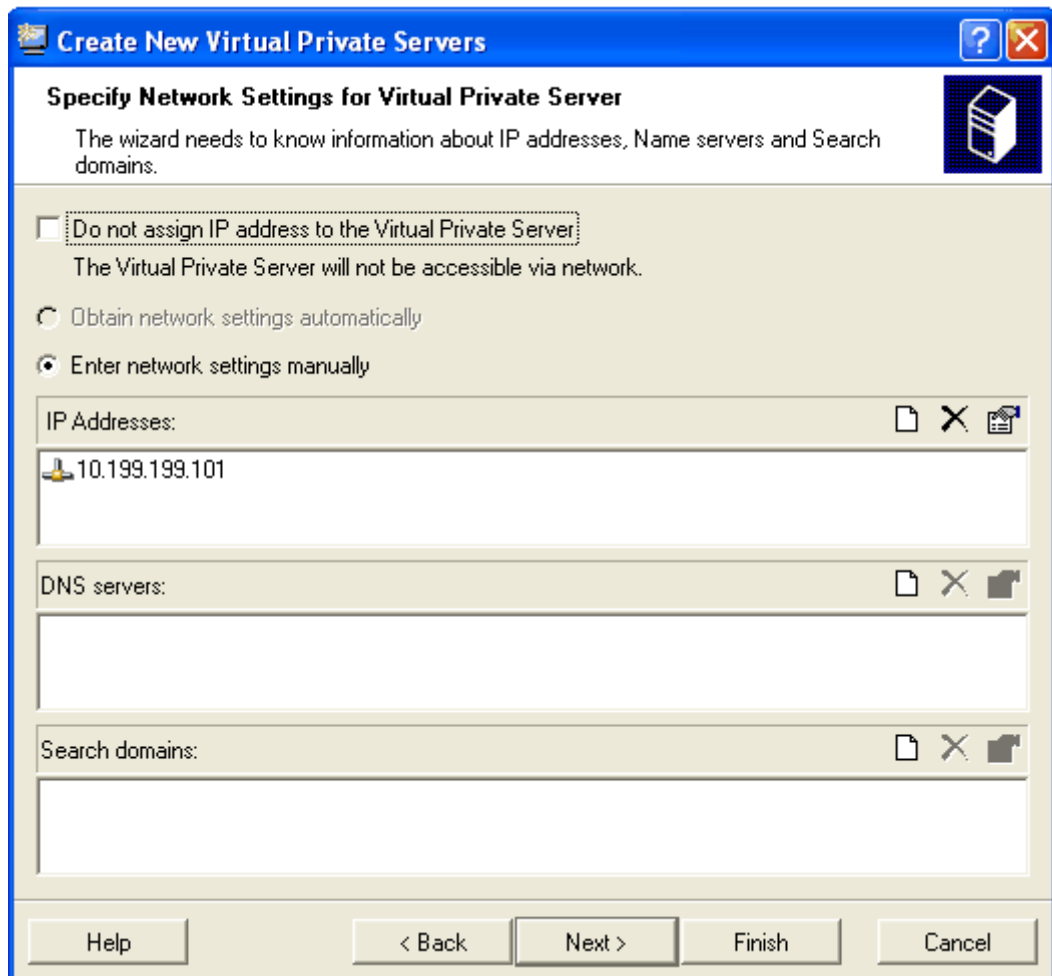


Figure 7: VZMC - Assigning IP Addresses When Creating Virtual Private Servers

- The **Assign IP addresses automatically** option tells the wizard to take the IP addresses information from the corresponding cluster pool.
- The **Assign IP addresses starting from** option lets you specify the starting IP address for the first VPS to be created. The IP addresses for the following VPSs will be calculated by the consecutive incrementation by 1.
- The **Set IP addresses manually** option lets you assign specific IP addresses separately for each Virtual Private Server.

All the other pages of the wizard are self-explanatory, so there is no need in dwelling upon them here in detail. You have the possibility to:

- Choose the OS template as the VPS base and the application templates to be added to the VPSs. Detailed information on OS and application templates is provided in the **Managing Templates** chapter (on page 75) and in the **Managing EZ Templates** chapter (on page 93), respectively.
- Change the default VPS private area and root paths or leave them intact.
- Configure Quality of Service and network shaping parameters. The Quality of Service and network shaping parameters are explained in the **Managing Resources** chapter. Please consult it to gather more understanding of these topics.
- Define what `iptables` modules are to be used inside the VPS. Detailed information on `iptables` is provided in the **Loading iptables Modules** section (on page 253).
- Specify whether the VPS is to be started after creation and whether offline management thru **Virtuozzo Power Panels** is enabled for the VPS(s). For information on the offline management feature, please see the **Setting VZCC/VZPP to Work** chapter in the **Virtuozzo Installation Guide**.

Creating a new Virtual Private Server may take some time. You can see the progress in the **Actions** pane.

After you have created, for example, a VPS with ID 102, you can see it in the right pane of the VZMC window:

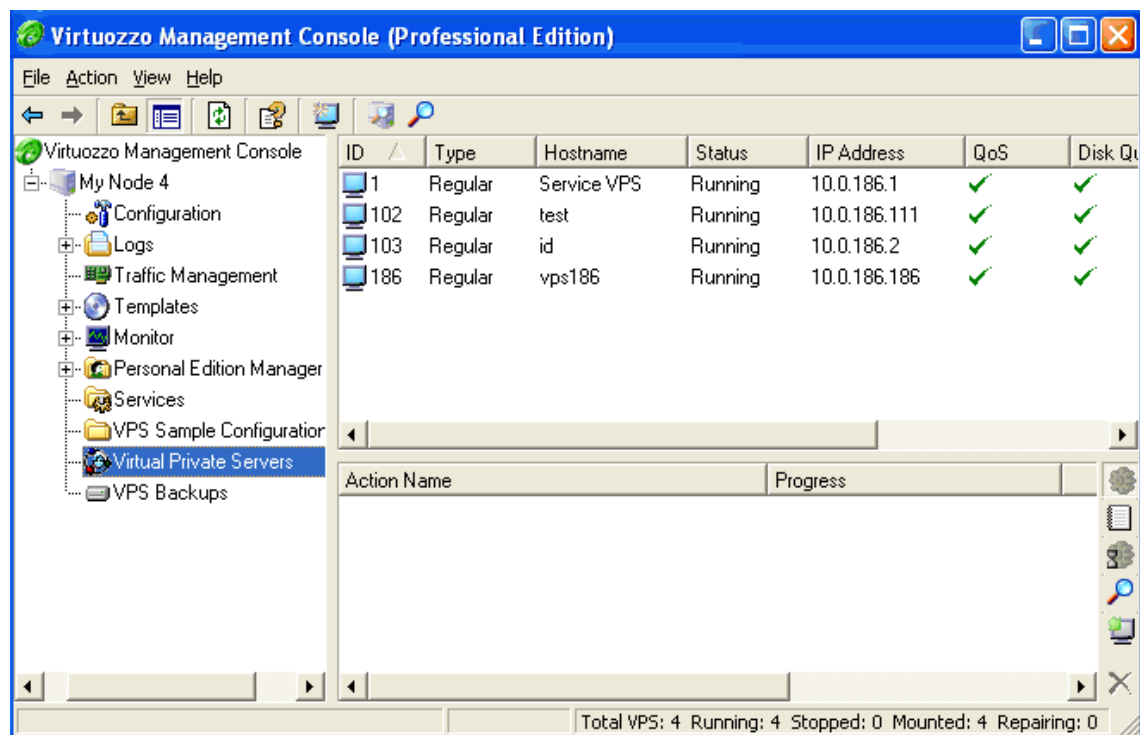


Figure 8: VZMC - Checking Newly-Created Virtual Private Server

Select the newly-created Virtual Private Server and choose the **Properties** item from the **Action** menu (or use the context menu, if you like). You will have the possibility to review and/or change most of the configuration options for this Virtual Private Server, as well as to set the root password using the **Advanced** tab.

Configuring Virtual Private Server

Configuring a Virtual Private Server consists of several tasks:

- Setting Virtual Private Server startup parameters;
- Setting Virtual Private Server network parameters;
- Setting Virtual Private Server user passwords;
- Configuring Quality of Service (Service Level) parameters.

For all these tasks, the `vzctl set` command is used. Using this command for setting VPS startup parameters, network parameters, and user passwords is explained later in this subsection. Service Level Management configuration topics are dwelled upon in the [Managing Resources](#) chapter.

Setting Startup Parameters

The `vzctl set` command allows you to define the `onboot` VPS startup parameter. Setting this parameter to `yes` makes your Virtual Private Server automatically boot at the Hardware Node startup. For example, to enable VPS 101 to automatically start on your Hardware Node boot, you can execute the following command:

```
# vzctl set 101 --onboot yes --save
Saved parameters for VPS 101
```

The `onboot` parameter will have effect only on the next VPS startup.

Setting Network Parameters

In order to be accessible from the network, a Virtual Private Server shall be assigned a correct IP address and hostname; DNS servers shall also be configured. In addition, the SSH or Telnet daemon shall be running inside the Virtual Private Server. The session below illustrates setting the Virtual Private Server 101 network parameters:

```
# vzctl set 101 --hostname test101.swsoft.com --save
Hostname for VPS set: test101.swsoft.com
Saved parameters for VPS 101
# vzctl set 101 --ipadd 10.0.186.1 --save
Adding IP address(es): 10.0.186.1
Saved parameters for VPS 101
# vzctl set 101 --nameserver 192.168.1.165 --save
File resolv.conf was modified
Saved parameters for VPS 101
```

This command will assign VPS 101 the IP address of 10.0.186.1, the hostname of test101.swsoft.com, and set the DNS server address to 192.168.1.165. The `--save` flag saves all the parameters to the VPS configuration file.

You can issue the above commands when the Virtual Private Server is running. In this case, if you do not want the applied values to persist, you can omit the `--save` option and the applied values will be valid only until the Virtual Private Server shutdown.

To check whether SSH is running inside the Virtual Private Server, use `vzctl exec`, which allows executing any commands in the Virtual Private Server context. In Red Hat 9 and other new OS templates, `sshd` is dependent on `xinetd`, so run:

```
# vzctl start 101
[This command starts VPS 101, if it is not started yet]
# vzctl exec 101 service xinetd status
xinetd is stopped
# vzctl exec 101 service xinetd start
Starting xinetd: [ OK ]
# vzctl exec 101 service xinetd status
xinetd (pid 16036) is running...
```

The above example assumes that VPS 101 is created on the Red Hat Linux template. For other OS templates, please consult the corresponding OS documentation.

For more information on running commands inside a VPS from the Hardware Node, see the [Running Commands in Virtual Private Server](#) subsection (on page 74).

Setting root Password for VPS

Setting the `root` user password is necessary for connecting to the Virtual Private Server via SSH or VZPP. By default, the `root` account is locked in a newly created VPS, and you cannot log in. In order to log in to the VPS, it is necessary to create a user account inside the Virtual Private Server and set a password for this account or unlock the root account. The easiest way of doing it is to run:

```
# vzctl start 101
[This command starts VPS 101, if it is not started yet]
# vzctl set 101 --userpasswd root:test
```

In this example, we set the root password for VPS 101 to “test”, and you can log in to the Virtual Private Server via SSH as root and administer it in the same way as you administer a standalone Linux computer: install additional software, add users, set up services, and so on. The password will be set inside the VPS in the `/etc/shadow` file in an encrypted form and will not be stored in the VPS configuration file. Therefore, if you forget the password, you have to reset it. Note that `--userpasswd` is the only option of the `vzctl set` command that never requires the `--save` switch, the password is anyway persistently set for the given Virtual Private Server.

While you can create users and set passwords for them using the `vzctl exec` or `vzctl set` commands, it is suggested that you delegate user management to the Virtual Private Server administrator advising him/her of the VPS `root` account password.

Starting, Stopping, Restarting, and Querying Status of Virtual Private Server

When a Virtual Private Server is created, it may be started up and shut down like an ordinary computer. To start Virtual Private Server 101, use the following command:

```
# vzctl start 101
Starting VPS ...
VPS is mounted
Adding port redirection to VPS(1): 4643 8443
Adding IP address(es): 10.163.163.101
Hostname for VPS 101 set: test.swsoft.com
VPS start in progress...
```

To check the status of a VPS, use the `vzctl status vps_id` command:

```
# vzctl status 101
VPSID 101 exist mounted running
```

Its output shows the following information:

- Whether the VPS private area exists;
- Whether this private area is mounted;
- Whether the Virtual Private Server is running.

In our case, `vzctl` reports that VPS 101 exists, its private area is mounted, and the VPS is running. Alternatively, you can make use of the `vzlist` utility:

```
# vzlist 101
VPSID      NPROC STATUS  IP_ADDR      HOSTNAME
101         20  running  10.0.186.101 test.swsoft.com
```

Still another way of getting the VPS status is checking the `/proc/vz/veinfo` file. This file lists all the Virtual Private Servers currently running on the Hardware Node. Each line presents a running Virtual Private Server in the `<VPS_ID> <VPS_class> <number_of_processes> <IP_address>` format:

```
# cat /proc/vz/veinfo
      101      2    20   10.0.186.1
       0      0    48
```

This output shows that VPS 101 is running, its class ID is “2”, i.e. unlimited, there are 20 running processes inside the VPS, and its IP address is 192.168.1.1. The second line corresponds to the VPS with ID 0, which is the Hardware Node itself.

The following command is used to stop a Virtual Private Server:

```
# vzctl stop 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
# vzctl status 101
VEID 101 exist unmounted down
```

`vzctl` has a two-minute timeout for the VPS shutdown scripts to be executed. If the VPS is not stopped in two minutes, the system forcibly kills all the processes in the Virtual Private Server. The Virtual Private Server will be stopped in any case, even if it is seriously damaged. To avoid waiting for two minutes in case of a Virtual Private Server that is known to be corrupt, you may use the `--fast` switch:

```
# vzctl stop 101 --fast
Stopping VPS ...
VPS was stopped
VPS is unmounted
```

Make sure that you do not use the `--fast` switch with healthy VPSs, unless necessary, as the forcible killing of VPS processes may be potentially dangerous.

The `vzctl start` and `vzctl stop` commands initiate the normal Linux OS startup or shutdown sequences inside the Virtual Private Server. In case of a Red Hat-like distribution, System V initialization scripts will be executed just like on an ordinary computer. You can customize startup scripts inside the Virtual Private Server as needed.

To restart a Virtual Private Server, you may as well use the `vzctl restart` command:

```
# vzctl restart 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
Starting VPS ...
VPS is mounted
Adding IP address(es): 10.0.186.101
VPS start in progress...
```

In VZMC, you can see the status of all the Virtual Private Servers of the given Hardware Node simply by selecting the **Virtual Private Servers** item under the corresponding Node.

There are two kinds of Virtual Private Servers in the **Virtual Private Servers** table. You may change the properties of and control those that are marked with grey monitor icons. The Virtual Private Servers marked with pink monitor icons provide critical services for Virtuozzo (for example, the Service VPS). You cannot change the properties of or stop such Virtual Private Servers.

To start or stop one or more Virtual Private Servers, select it (them) in the **Virtual Private Servers** table in the right pane. You can use **CTRL+Click** to select or deselect an entry, **SHIFT+Click** to select a range of Virtual Private Servers, **CTRL+A** to select all Virtual Private Servers. Then press the **Start** or **Stop** button on the toolbar or select **Start** or **Stop** from the **Action** menu.

Starting or stopping a considerable number of Virtual Private Servers may take a rather long run. The progress is displayed in the **Actions** pane.

Listing Virtual Private Servers

Very often you may want to get an overview of the Virtual Private Servers existing on the given Hardware Node and to get additional information about them - their IP addresses, hostnames, current resource consumption, etc. In the most general case, you may get a list of all VPSs by issuing the following command:

```
# vzlist -a
      VPSID      NPROC STATUS  IP_ADDR      HOSTNAME
      1          135 running 10.101.60.79  localhost
      101         8 running 10.101.66.1   vps101.eng.sw.ru
      102         7 running 10.101.66.159 vps102.eng.sw.ru
      103         - stopped 10.101.66.103 vps103.eng.sw.ru
```

The `-a` switch tells the `vzlist` utility to output both running and stopped Virtual Private Servers. By default, only running VPSs are shown. The default columns inform you of the VPS IDs, the number of running processes inside VPSs, their status, IP addresses, and hostnames. This output may be customized as desired by using `vzlist` command line switches. For example:

```
# vzlist -o veid,diskinodes.s -s diskinodes.s
      VPSID DQINODES.S
      1      400000
      101     200000
      102     200000
```

This shows only running VPSs with the information about their IDs and soft limit on disk inodes (see the [Managing Resources](#) chapter for more information), with the list sorted by this soft limit. The full list of the `vzlist` command line switches and output and sorting options is available in the `vzlist` subsection of the [Virtuozzo Command Line Interface](#) chapter in the [Virtuozzo Reference Guide](#).

Very often you may want to get an overview of the Virtual Private Servers existing on the given Hardware Node and to get additional information about them - their IP addresses, hostnames, status, etc. In VZMC, you may display a list of all VPSs by clicking the [Virtual Private Servers](#) item:

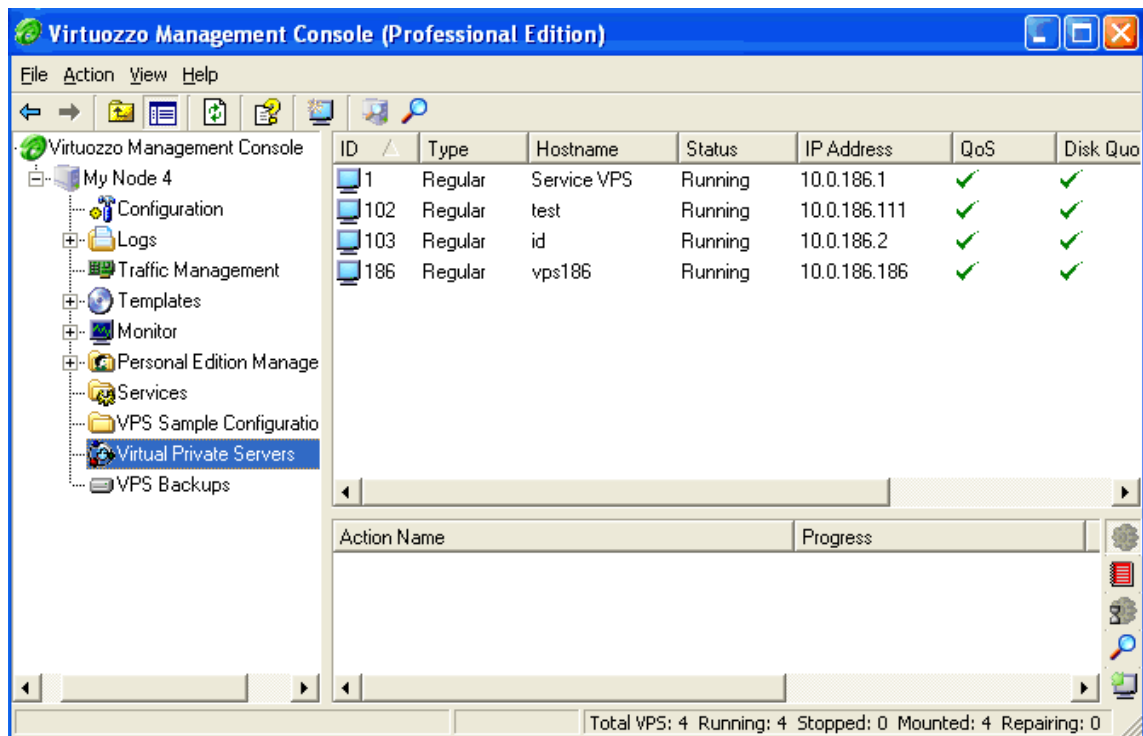


Figure 9: VZMC - Viewing List of VPSs

You can see that currently the Service VPS with ID 1 and some regular VPSs exist on the Hardware Node. To facilitate working with VPSs, you can sort them by different parameters: their ID, type, hostname, status, IP address, etc. Just click the column with the appropriate name to put VPSs in the desired order.

Migrating Virtual Private Server

The Virtuozzo Hardware Node is the system with higher availability requirements in comparison with a typical Linux system. If you are running your company mail server, file server, and web server in different Virtual Private Servers on one and the same Hardware Node, then shutting it down for hardware upgrade will make all these services unavailable at once. To facilitate hardware upgrades and load balancing between several Hardware Nodes, Virtuozzo provides you with the ability to migrate Virtual Private Servers from one physical box to another.

Migrating VPSs is possible if Virtuozzo for Linux is installed on two or more Hardware Nodes, so you are able to move a Virtual Private Server to another Node. Migration may be necessary if a Hardware Node is undergoing a planned maintenance or in certain other cases. In Virtuozzo 3.0, you can choose one of the following ways to migrate a VPS:

- Migrating a Virtual Private Server by using the standard migration technology. In this case there is a short downtime needed to stop and start the VPS during its migration from the Source Node to the Destination Node.
- Migrating a Virtual Private Server by using the zero downtime migration technology. In this case the 'stop' and 'start' operations are not performed and the migrated Virtual Private Server is restored on the Destination Node in the same state as it was at the beginning of the migration. This greatly reduces the migration time and puts it on the same footing as the delay caused by a short interruption in the network connectivity.

Both ways are described in the following subsections in detail.

Standard Migration

The standard migration procedure allows you to move both stopped and running Virtual Private Servers. Migrating a stopped VPS includes copying all VPS private files from one Node to another and does not differ from copying a number of files from one computer to another over the network. In its turn, the migration procedure of a running VPS is a bit more complicated and may be described as follows:

- 1 After initiating the migration process, all VPS private data are copied to the Destination Node. During this time, the VPS on the Source Node continues running.
- 2 The VPS on the Source Node is stopped.
- 3 The VPS private data copied to the Destination Node are compared with those on the Source Node and, if any files were changed during the 1st migration step, they are copied to the Destination Node again and rewrite the outdated versions.
- 4 The VPS on the Destination Node is started.

There is a short downtime needed to stop the VPS on the Source Node, copy the VPS private data changes to the Destination Node, and start the VPS on the Destination Node. However, this time is very short and does not usually exceed one minute.

Note: Before the migration, it might be necessary to detach the Virtual Private Server from its caches. For more information on cached files, see the [Cleaning Up Virtual Private Servers](#) subsection (on page 124).

The following session moves VPS 101 from the current Hardware Node to a new one named `ts7.swsoft.com`:

```
# vzmigrate ts7.swsoft.com 101
root@ts7.swsoft.com's password:
vzmsrc: Connection to destination HN (ts7.swsoft.com) \
is successfully established
vzmsrc: Moving/copying VE#101 -> VE#101, [], [] ...
vzmsrc: VE migrating mode : first stage sync, with tracking, \
second stage sync, with VE stopping
vzmsrc: Syncing private area of VE#101 [/vz/private/101] ...
/ 100% |*****|
vzmsrc: done
vzmsrc: Stopping VE#101 ...
vzmsrc: done
vzmsrc: Fast syncing private area of VE#101 [/vz/private/101] ...
/ 100% |*****|
vzmsrc: done
vzmsrc: DST: Starting VE#101 ...
vzmsrc: DST: done
vzmsrc: Successfully completed
```

You can specify more than one VPS ID simultaneously; in this case, all specified Virtual Private Servers will be moved to a new Hardware Node one by one.

By default, after the migration process is completed, the private area is removed from the original Hardware Node, and the VPS configuration file is renamed by receiving the `.migrated` suffix. However, if you prefer to keep the private area on the source Node as well, you can override the default behavior by changing the value of the `REMOVEMIGRATED` variable in the VirtuoZZo global configuration file (`/etc/sysconfig/vz`) to “no” or by using the `-r` switch of the `vzmigrate` command.

To migrate one or more Virtual Private Servers to another Hardware Node with VirtuoZZo for Linux using VZMC, select these Virtual Private Servers from the list in the right pane after selecting the **Virtual Private Servers** item in the left pane. Then right-click the selection and point to **Tasks -> Migrate to Another Hardware Node** on the context menu. Note that the target Hardware Node must be already registered in VZMC; otherwise, the migration option will not be available. A migration dialog appears, for example:

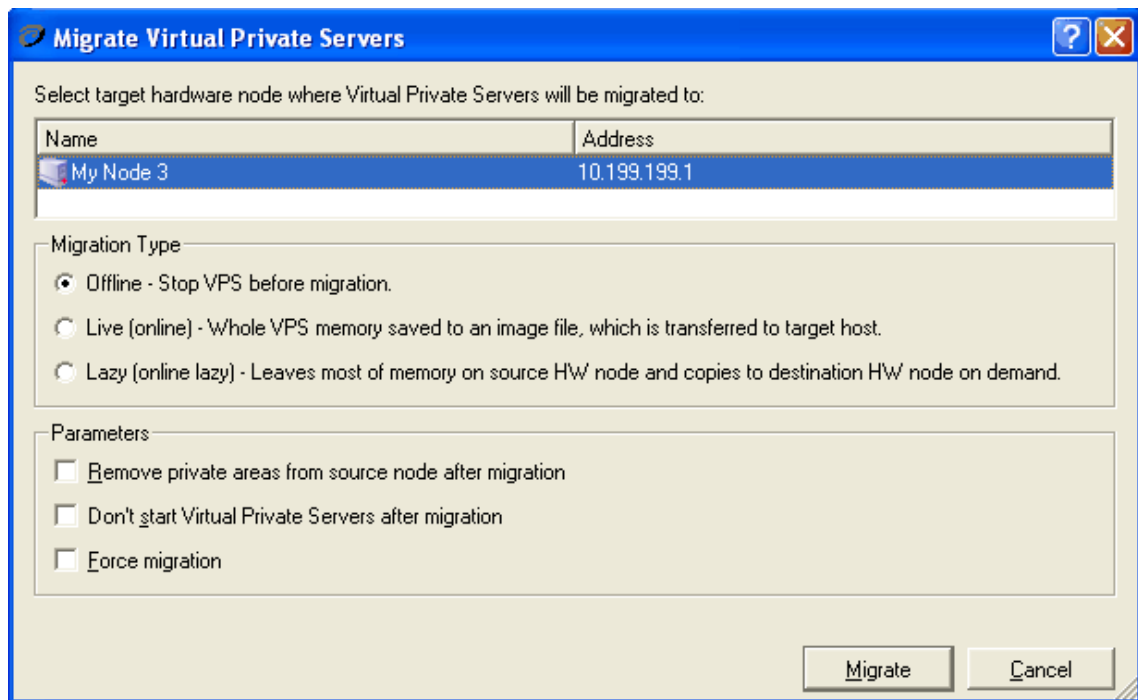


Figure 10: VZMC - Migrating VPSs

In this window, you should select the **Offline - Stop VPS before migration** radio button under the **Migration Type** group to migrate the VPS by using the standard migration technology. Under the **Parameters** group, you can also choose one of the following options:

- Select the **Remove private areas ...** checkbox to delete the VPS private area from the Source Node after the VPS successful migration.
- The **Don't start Virtual Private Servers after migration** checkbox, when selected, prevents the migrated VPS from starting on the Destination Node after its successful migration. This option does not have any effect if the Virtual Private Server was not running on the Source Node.
- The **Force migration** checkbox, when selected, forces the VPS migration even if the templates necessary for the VPS correct operation are not installed on the Destination Node. However, it will be impossible to start such a VPS after the migration in case of the absence of the needed templates.

Finally, you should select the Destination Node where you want to migrate the selected Virtual Private Server(s) and press the **Migrate** button.

Zero-Downtime Migration

Starting with Virtuozzo 3.0, the `vzmigrate` utility allows you to migrate your Virtual Private Servers from one Hardware Node to another with zero downtime. The zero downtime migration technology has the following advantages as compared with the standard one:

- The process of migrating a VPS to another Node is transparent for you and the VPS applications and network connections, i.e., on the Source and Destination Nodes, no modifications of system characteristics and operational procedures inside the VPS are performed.
- The VPS migration time is greatly reduced (in fact, the migration is almost invisible for the VPS user) and can be compared with the delay caused by a short interruption in the network connectivity.
- The Virtual Private Server is restored on the Destination Node in the same state as it was at the beginning of the migration.
- You can move the Virtual Private Servers running a number of applications which you do not want to be rebooted during the migration for some reason or another.

Notes: 1. Zero-downtime migration cannot be performed on Virtual Private Servers with bind mounts inside them.

2. Zero-downtime migration can be performed only on Virtual Private Servers running the Linux distributions with the 2.6 kernel.

Before performing zero-downtime migration, it is recommended to synchronize the system time on the Source and Destination Nodes, e.g. by means of NTP (<http://www.ntp.org>). The reason for this recommendation is that some processes running in the VPS might rely on the system time being monotonic and thus might behave unpredictably if they see an abrupt step forward or backward in the time once they find themselves on the new Node with different system clock parameters.

To migrate a VPS by using the zero downtime migration technology, you should pass the `--online` option to the `vzmigrate` utility. For example, you can move VPS 101 from the current Hardware Node to the Destination Node named `my_node.com` by executing the following command:

```
# vz migrate --online my_node.com 101
Enter password:
Connection to destination HN (192.168.1.57) \
is successfully established
Moving/copying VPS#101 -> VPS#101, [], [] ...
Syncing private area '/vz/private/101'
- 100% |*****
done
Suspending VPS#101 ...
done
Dumping VPS#101 ...
done
...
Migration completed
```

You can specify more than one VPS ID simultaneously; in this case, all specified Virtual Private Servers will be moved to a new Hardware Node one by one.

If you have a Virtual Private Server with the high memory consumption, you can additionally speed up the migration process by passing the `--lazy` option to `vzmigrate`. This makes `vzmigrate` decrease the size of the image file storing all VPS private data and transferred to the Destination Node by leaving the main amount of memory in a locked state on the Source Node and swapping this memory from the Source Node on demand. Thus, the migrated Virtual Private Server can be started before the whole memory is transferred to the Destination Node, which drastically reduces the service delay of the corresponding VPS. When a process tries to access a page of memory that has not yet been migrated, the request is intercepted and redirected to the Source Node where this page is stored.

Note: If the CPU capabilities on the Source Node exceed those on the Destination Node (e.g. you migrate from a Source Node running the Pentium 4 processor to a Destination Node running the Pentium 3 processor), the migration may fail and you will be presented with the corresponding warning message. However, if you are sure that the CPU power on the Destination Node is sufficient to start and run the VPS(s) being migrated, you can use the `-f` option to force the migration process.

To migrate one or more Virtual Private Servers in VZMC, select these Virtual Private Servers from the list in the right pane after selecting the **Virtual Private Servers** item in the left pane. Then right-click the selection and point to **Tasks -> Migrate to Another Hardware Node** on the context menu. Note that the target Hardware Node must be already registered in VZMC; otherwise, the migration option will not be available. A migration dialog appears, for example:

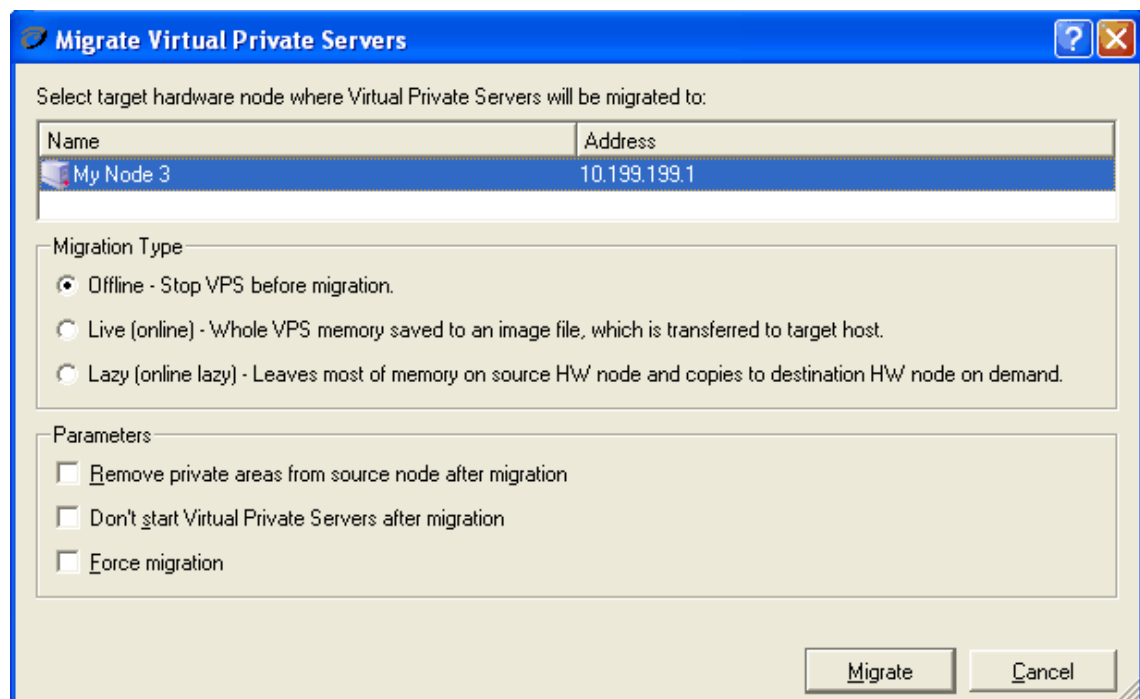


Figure 11: VZMC - Migrating VPSs

In this window, you should do the following:

- Under the **Migration Type** group, select either the **Live (online) ...** or the **Lazy (online lazy) ...** radio button to migrate the Virtual Private Server by using the zero downtime migration technology. The latter enables you to additionally speed up the migration process and can be efficiently used if your VPS is running a number of memory-consuming applications.

- Under the **Parameters** group, review the three checkboxes and check up the needed ones:
 - Select the **Remove private areas ...** checkbox to delete the VPS private area from the Source Node after successful migration.
 - The **Don't start Virtual Private Servers after migration** checkbox, when selected, prevents the migrated VPS from starting on the Destination Node after its successful migration. This option does not have any effect if the Virtual Private Server was not running on the Source Node.
 - The **Force migration** checkbox, when selected, forces the VPS migration even if the templates necessary for the VPS correct operation are not installed on the Destination Node. However, it will be impossible to start such a VPS after the migration in case of the absence of the needed templates.
- Select the target Hardware Node where you want to migrate the selected Virtual Private Server(s) and press the **Migrate** button.

Backing Up and Restoring Virtual Private Servers

A regular backing up of the existing Virtual Private Servers is essential for any Hardware Node reliability. It is highly recommended to set up a dedicated Node for running the backing-up utilities and storing the VPS backups (further on, the Backup Node). In a more general case, any Hardware Node can be assigned an additional role of the Backup Node.

Any Virtual Private Server is defined by its private area, configuration files, action scripts, and quota information. Virtuozzo allows to back up all these components. Each backup file may be of one of the following 3 types:

- a plain full backup;
- a full backup differing from a plain full backup in that it contains additional auxiliary information allowing further incremental backups;
- an incremental backup containing only the files changed since the full backup or the previous incremental backup.

An incremental backup may prove very useful because it records only the changes since the last VPS backup and therefore is much less in size and takes much less time than the full backup. However, after several consecutive incremental backups it is recommended to create a full backup de nouveau and start the incremental backups chain from scratch.

If an incremental backup is performed, and the corresponding full backup cannot be found, a full backup is automatically performed.

Running Backup Utilities on Backup Node

The `vzbackup` utility is run on the Backup Node connecting via SSH to the Hardware Node(s) where some or all Virtual Private Servers are to be backed up and puts the tarballs (either compressed or not) into the directory defined in the `/etc/vzbackup.conf` global backup configuration file (by default, this directory is `/vz/backups`). Later on, the VPS backups may be restored from this directory. Assuming that you are going to back up all the VPSs on Node 192.168.1.165, you may run the following session on the Backup Node:

```
# vzbackup -i -Cg -s 192.168.1.165
root@192.168.1.165's password:
vzbackup(16013): Starting backup. Nodes - dhcp-165.asplinux.ru.
vzbackup(16013): Starting node dhcp-165.asplinux.ru backup...
vzbackup(16013): Node dhcp-165.asplinux.ru archived \
VPSs: 101(2003-06-27T115742+0400@dhcp-165.asplinux.ru) \
1(2003-06-27T115749+0400@dhcp-165.asplinux.ru)
vzbackup(16013): Cleaning up
```

The `-s` option in the session above forces the VPSs to be stopped for the time necessary for their backing up. In this case, if a client tries to access the VPSs during their downtime, a temporary "busy" page is shown. You may use the `-n` option to back up the VPSs while they are running, which is better, on the one hand, but does not guarantee the VPSs consistency, on the other. If neither `-n` nor `-s` is specified, the backup configuration file is consulted for the default value (the `BACKUP_VESTOP` parameter).

The `-Cg` switch compresses the resulting VPS backups with the `gzip` archiver. You can also use the `bzip2` algorithm or do without any compression. Whether to use compression or not and the compression method itself are defined in the global backup configuration file and can be overridden by the command line switches. For a full list of configuration file parameters and command line options, turn to the [Virtuozzo Reference Guide](#).

Note: You may as well compress the VPS tarballs manually later. Provided the archives will be situated in the same directory and will have an identical name (before the extension), this will not hinder the VPSs from being restored when time comes. A VPS may be restored from files with any of the following extensions: `.tar`, `.tar.gz`, `.tgz`, `tar.bz2`.

The type of the backup to be performed is specified by one of the following 3 options:

- `-F` a plain full backup;
- `-I` a full backup;
- `-i` an incremental backup containing only the files changed since the previous "I" or "i" backup.

If you specify the `-i` option, and the utility cannot find the corresponding full backup, a full backup is performed.

You may specify any number of Hardware Nodes names or IP addresses in the command line. You may also enter these names as the value of the `BACKUP_NODES` parameter in the global backup configuration file to avoid the necessity to specify them in the command line. In this case, you shall specify the `-a` option instead.

If you wish to back up not all, but select Virtual Private Servers from the specified Node, use the `-e` or `-x` switches (include or exclude the specified VPSs, respectively). For example:

```
# vzbackup -i -s 192.168.1.165 -e 101
root@192.168.1.165's password:
vzbackup(17344): Starting backup. Nodes - dhcp-165.asplinux.ru.
vzbackup(17344): Starting node dhcp-165.asplinux.ru backup...
vzbackup(17344): Node dhcp-165.asplinux.ru archived VPSs: \
101(2003-06-27T120326+0400@dhcp-165.asplinux.ru)
vzbackup(17344): Cleaning up...
```

In this session, only VPS 101 will be included in the backup.

Note: A number of default parameters in the global backup configuration file may be adjusted for a particular Hardware Node to be backed up. To this effect, you should create a new configuration file named `<node>.conf` and put it to the backup directory (defined by the `BACKUP_DIR` parameter in the global backup configuration file). This file should contain those parameters that you want to re-write for a particular Node. Still, a number of other parameters may further be adjusted for a particular VPS to be backed up. In this case these parameters should be inserted into the corresponding VPS configuration file (`/etc/sysconfig/vz-scripts/vpsid.conf`). For a complete list of those backup parameters that are allowed to be used in per-Node and per-VPS configuration files, see the **Backup Configuration File** section in the **Virtuozzo Reference Guide**.

To restore any Hardware Nodes previously backed up or separate Virtual Private Servers, you might want to view first the information about these VPSs:

```
# vzrestore -l
VE 101 dir /vz/backups/dhcp-165.asplinux.ru
2003-06-27T122705+0400@dhcp-165.asplinux.ru type I
```

To do the restoring proper, issue the following command:

```
# vzrestore dhcp-165.asplinux.ru -e 101
vzrestore(23558): Starting restore VPS \
101(2003-06-27T122705+0400@dhcp-165.asplinux.ru) \
on node dhcp-165.asplinux.ru...
vzrestore1 (101): VPS is mounted. Exiting...
vzrestore(23558): Failed to restore dhcp-165.asplinux.ru VPS 101 tag .
vzrestore(23558): Done.
# vzctl stop 101
Removing stale lock file /vz/lock/101.lck
Stopping VPS ...
VPS was stopped
VPS is unmounted
# vzrestore dhcp-165.asplinux.ru -e 101
vzrestore(23960): Starting restore VPS \
101(2003-06-27T122705+0400@dhcp-165.asplinux.ru) \
on node dhcp-165.asplinux.ru...
vzrestore1 (101): Created /vz/backup/101/tmpfyPsqw
vzrestore1 (101): Restoring VPS...
vzrestore1 (101): \
/vz/backups/dhcp-165.asplinux.ru/101/101-2003-06-27T122705+ \
0400@dhcp-165.asplinux.ru-I.tar...
vzrestore1 (101): Replacing /etc/sysconfig/vz-scripts/101.conf...
vzrestore1 (101): Moving /vz/private/101 to \
vz/backup/101/tmpfyPsqw/101.tmp...
Saved parameters for VPS 101
vzrestore1 (101): Cleaning up...
vzrestore1 (101): Done.
vzrestore(23960): Done.
```

Use the `-e` or `-x` switches in the same way as for the `vzbackup` utility. You may also use the `-d` option to specify the Destination Node where the VPSs are to be restored instead of the default behaviour of restoring the VPSs to their Source Node.

Note: The `vzrestore` utility does not restore VPSs by the Hardware Nodes IP addresses, but only by their hostnames, even if the backing up was performed by means of IP addresses.

Managing Backups in VZMC

VZMC deals with three kinds of Nodes - the *Source Nodes* (the Nodes where VPSs are hosted during their backing up); the *Backup Nodes* (the Nodes where VPS backups are stored); and the *Destination Nodes* (the Nodes where VPS backups are restored). These Nodes are singled out by their functionality only. In reality, one and the same Hardware Node may perform two or even three functions. Usually, the Source and Destination Node are represented by one and the same HN, because you will likely want the VPSs you back up to be restored to their original Node. However, setting up a dedicated Backup Node is recommended.

You should make sure that all the three Nodes are registered in VZMC before starting to work with them.

VZMC lets you perform the following backup-related functions:

- Assigning the default Backup Node for the given Source Node;
- Back up a single VPS from the Source Node to the Backup Node;
- Back up a number of VPSs or the whole Hardware Node (i.e. all the VPSs on the given Node) to the Backup Node;
- Restore a single VPS from the Backup Node to the Destination Node;
- Restore a number of VPSs or the whole Hardware Node from the Backup Node;
- Directly manage the Backup Nodes;
- Search the backup of a given VPS from the Source Node across all the Backup Nodes.

Assigning Default Backup Node

When you are backing up Virtual Private Servers from a Source Node, you shall always specify on what Node the resulting backups should be placed, i.e. the Backup Node. The **Default Backup Node** window allows you to assign the default Backup Node for the given Source Node, i.e. for the Node for which the window has been invoked. To display this window, right-click the respective Source Node and choose **Backup --> Set Default Backup Node** on the context menu.

The window presents a list of Nodes registered in VZMC together with their IP addresses. If the default Backup Node already exists for the given Source Node, it is selected in the table. You should select the Node you wish to be the default Backup Node for the given Source Node and press **OK**. The assignment of the default Backup Node brings about the following effects:

- When backing up Virtual Private Servers from the corresponding Source Node, you will be automatically suggested to place the backups onto the default Backup Node. This facilitates keeping track of those places where various backups are stored.
- When a VPS administrator backs up their Virtual Private Server by means of Virtuozzo Power Panels (VZPP) or Virtuozzo Control Center (VZCC), the corresponding backup is automatically placed on the default Backup Node.

There are no restrictions as to what Hardware Node may be the default Backup Node; it may even be the Source Node itself. In the latter case, the VPS backups will be placed on the same Node where the corresponding Virtual Private Servers are hosted. The only thing you have to ensure is that the default Backup Node has sufficient disk space for housing multiple backups.

Backing Up Single VPS

To back up a Virtual Private Server from the Source Node to the Backup Node, do the following:

- 1** Expand the *Source Node* item in the left pane of the VZMC main window and click the **Virtual Private Servers** item to open the Virtual Private Server manager window.
- 2** Select the Virtual Private Server you wish to back up. You may use the CTRL and SHIFT keys for selecting a number of VPSs.
- 3** Click the right mouse button and select **Backup --> Back Up Virtual Private Server** on the context menu.

The Back Up Virtual Private Server wizard opens:

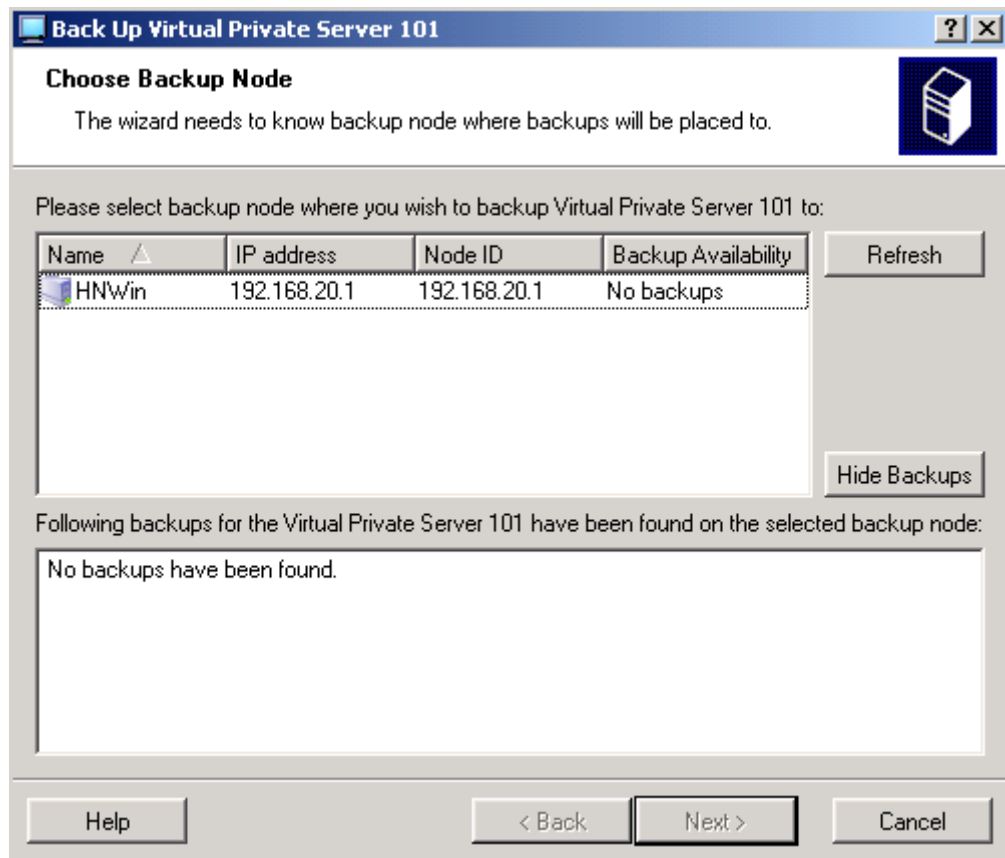


Figure 12: VZMC - Backing Up VPSs Wizard

In this wizard you should:

- Select the Backup Node. This Node is the place where the VPS backups will be stored. The only requirement for this Node is that it should be registered in VZMC; it may even be the Source Node itself. Often, it will be a dedicated Backup Node.
- In the next window, specify the backup type. It may be full or incremental. Detailed information on backup types is provided in the **Backing Up and Restoring Virtual Private Servers** section. If you are backing up a single VPS, and no backup of this VPS has been found on the Backup Node, this step is omitted, and a full backup is automatically created.
- On the Review Virtual Private Server Backup Settings page:
 - Stop the VPS(s) for the time necessary for their backing up or leave them running. Selecting the **Stop Virtual Private Server before backup** option forces the VPS(s) to be stopped. In this case, if a client tries to access the VPSs during their downtime, a temporary "busy" page is shown. If this checkbox is cleared, the VPSs are backed up while they are running, which is better, on the one hand, but does not guarantee the VPSs consistency, on the other.
 - Do not stop the VPSs backup even if any errors appear (the **Don't stop on errors** checkbox is selected) or break the backup process should any malfunction occur (the checkbox is cleared).

Backing Up Group of VPSs

To back up several or all VPSs from a single Source Node, right-click the **Virtual Private Servers** item under the corresponding *Source Node* and select **Tasks --> Back up Virtual Private Servers** on the pop-up menu. The **Back Up Virtual Private Servers** wizard is displayed. In this wizard you should:

- Select the Virtual Private Servers from the Source Node you wish to back up. By default, all the VPSs are selected, but you may exclude certain VPSs from the backup.
- On the next page of the wizard, select the Backup Node. This Node is the place where the VPS backups will be stored. The only requirement for this Node is that it should be registered in VZMC; otherwise it will not be displayed on this page. It may even be the Source Node itself. Often, it will be a dedicated Backup Node.
- On the Review page:
 - Stop the VPS(s) for the time necessary for their backing up or leave them running. Selecting the **Stop Virtual Private Server before backup** option forces the VPS(s) to be stopped. In this case, if a client tries to access the VPSs during their downtime, a temporary "busy" page is shown. If this checkbox is cleared, the VPSs are backed up while they are running, which is better, on the one hand, but does not guarantee the VPSs consistency, on the other.
 - Do not stop the VPSs backup even if any errors appear (the **Don't stop on errors** checkbox is selected) or break the backup process should any malfunction occur (the checkbox is cleared).
 - Force a full backup for all the VPSs (the **Force full backup** checkbox on the Review page is selected), or allow an incremental backup for those VPSs whose full backups are already present on the Backup Node (the checkbox is cleared).

Another way of backing up a number of VPSs from the given Source Node is the following:

- 1 Expand the *Source Node* item in the left pane of the VZMC main window and click the **Virtual Private Servers** item to open the Virtual Private Servers manager window.
- 2 Select the Virtual Private Servers you wish to back up. Use the CTRL and SHIFT keys for selecting a number of VPSs.
- 3 Click the right mouse button and select **Back up Virtual Private Servers** on the context menu.

The aforementioned **Back Up Virtual Private Servers** wizard is opened directly at the second page, because the first page (**Choose Virtual Private Servers to Back Up**) becomes unnecessary.

Restoring Single VPS

To restore a Virtual Private Server from its backup, do the following:

- 1 Expand the *Source Node* item in the left pane of the VZMC main window and click the **Virtual Private Servers** item to open the Virtual Private Servers manager window.
- 2 Select the Virtual Private Server the backup of which you wish to restore from the Backup Node.
- 3 Click the right mouse button and select **Backup --> Restore Virtual Private Server** on the context menu.

The Restore Virtual Private Server wizard opens:

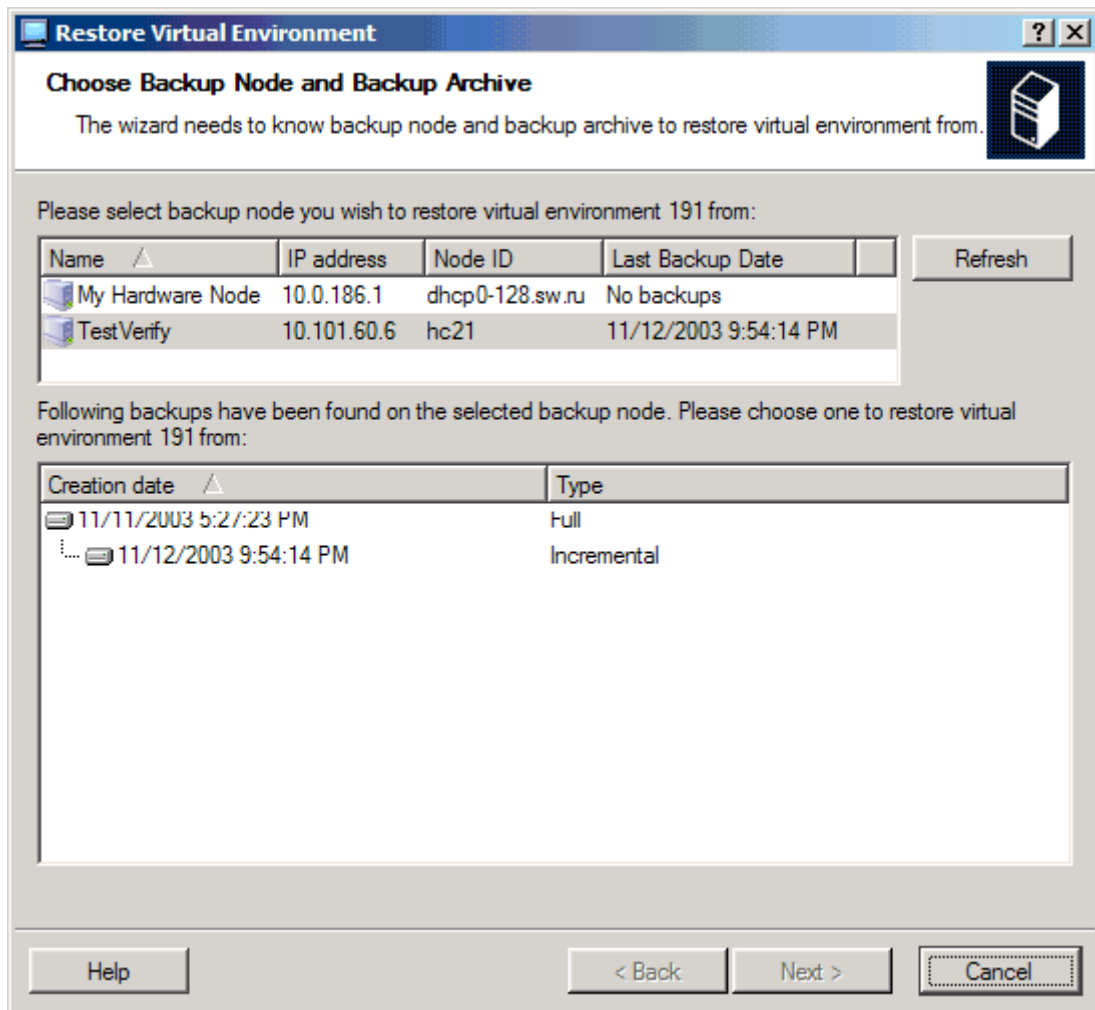


Figure 13: VZMC - Restoring VPSs Wizard

In this wizard you should:

- Select the Backup Node. This Node is the place where the VPS backup is stored. The **Last Backup Date** column in the list of Backup Nodes shows the date and time of the last backup (if any) of the selected VPS on the corresponding Node.
- Select the backup from which the VPS is to be restored. Any VPS may have any number of its backups made at different dates and of different types. As a rule, you choose the most recent backup, unless you have reasons to restore an intermediary one.

Note: During this operation, the Destination Node is supposed to be the same as the Source Node. For instructions on how to restore a VPS to a Destination Node other than the Source Node, see [Managing Backup Node](#).

Restoring Group of VPSs

To restore several Virtual Private Servers of a single Source Node from their backups on the Backup Node, right-click the **Virtual Private Servers** item under the corresponding *Source Node*, and select **Tasks --> Restore Virtual Private Servers** on the context menu. The **Restore Virtual Private Servers** wizard is displayed. In this wizard you should:

- Select the Backup Node. This Node is the place where the backups of the Source Node VPSs are stored. The **Backup Availability** column in the list of Backup Nodes shows whether backups have been found on the corresponding Node.
- Select the Virtual Private Servers you wish to restore from the Backup Node. By default, all the backups of the VPSs originally belonging to the Source Node are selected, but you may exclude certain VPSs from this list, as well as include in it any other backups found on this Backup Node (i.e. the backups of those VPSs not belonging to the Source Node). To include these other backups, you should first make them visible by selecting the **Show all available backups** checkbox.

Note: During this operation, all the VPSs will be restored to the Source Node, i.e. to the Node for which you have invoked the wizard, irrespective of whether the backed up Virtual Private Servers originally belonged to this Source Node or to any other Node.

Managing Backup Node

Any Hardware Node may perform the functions of the Backup Node, i.e. store the backups of any VPSs of any Hardware Nodes. To see a list of VPS backups stored on a Hardware Node, expand its name in the left pane of the VZMC main window and select the **VPS Backups** item:

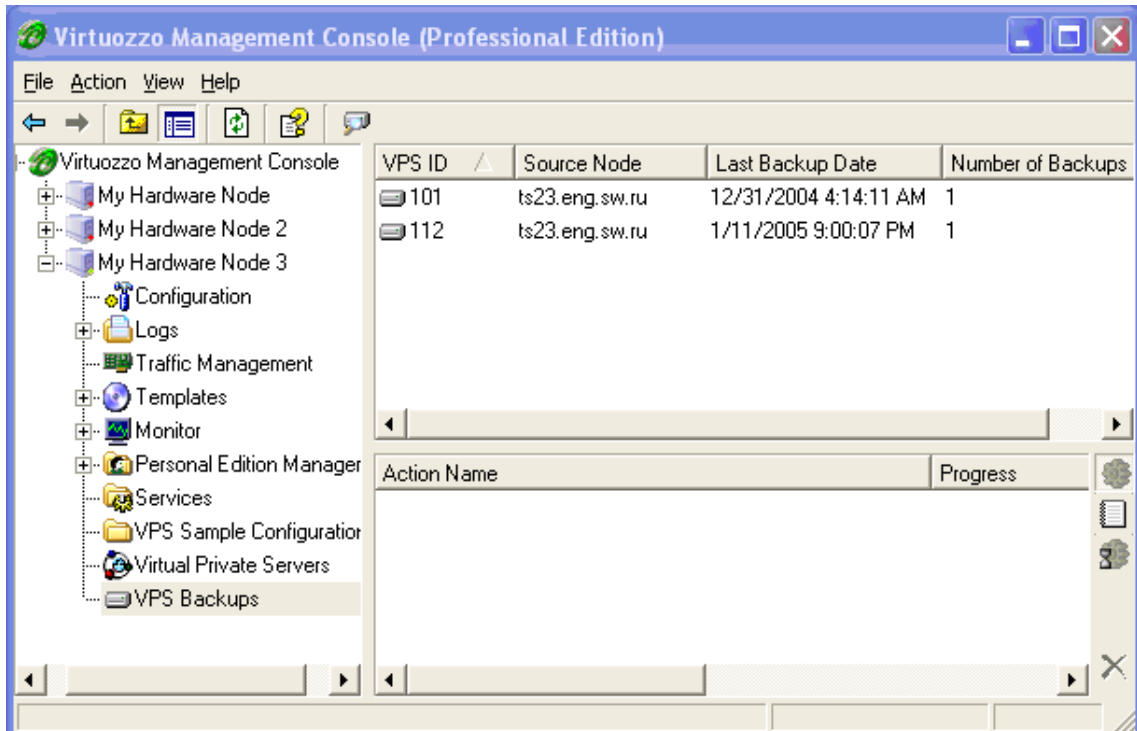


Figure 14: VZMC - Listing Backups

The table in the right pane presents the following information about the VPS backups stored on the current Backup Node:

Column Name	Description
VPS ID	The ID of the backed up Virtual Private Server.
Source Node	The Node where the VPS was hosted during its backing up.
Last Backup Date	The date and time when the last backing up of the VPS took place.
Number of Backups	The number of VPS backups on the Node.

You are able to restore a single VPS from its backup. Right-clicking the needed VPS backup and selecting **Restore Virtual Private Server** on the menu starts the **Restore Virtual Private Server** wizard. In this wizard you should:

- Select the Destination Node, i.e. the place whither the VPS will be restored. By default, the VPS Source Node is selected. Only the Nodes registered in VZMC are shown.
- Select the backup from which the VPS is to be restored. Any VPS may have any number of its backups made at different dates and of different types. As a rule, you choose the most recent backup, unless you have reasons to restore an intermediary one.

Right-clicking on a VPS backup in this table and selecting **Properties** on the context menu brings about the **Virtual Private Server Backups** dialog where you can view extensive information about the current VPS backup, including all its full and incremental backups, as well as delete any of these backups or restore them in the manner depicted above.

Searching for VPS Backups

If you do not remember the place where you are storing the backup of a particular VPS (identified by its ID or its IP address or its hostname or by the date of its creation), you can search for the backup across all the Hardware Nodes (performing the function of Backup Nodes in this case) registered in VZMC.

To search for a backup, do the following:

- 1 Right-click the **Virtual Private Servers** item under the corresponding *Source Node*, and select **Backup --> Search for Backups** on the pop-up menu to open the **Find Virtual Private Server Backups** dialog:

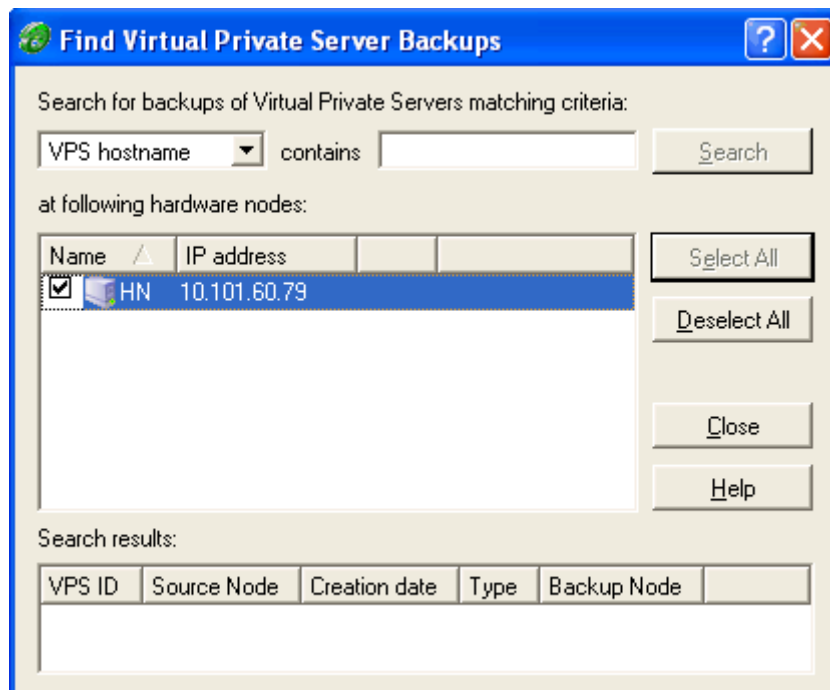


Figure 15: VZMC - Searching for Backups

- 2 On the upper left drop-down menu, choose the VPS parameter by which you wish to search for the corresponding VPS backup.

- 3 Enter the value of the parameter in the text field on the right. All the VPSs with the corresponding parameter including the specified value as its part will be found. E.g., if you enter "100" as the value for VPS ID, the backups of VPSs 100, 1000, 1001, 1002, 2100, 3100, and so on, will be searched for.
- 4 Check those Nodes where you want to search for the backups.
- 5 Press the **Search** button.

The **Search results** table presents the following information about the found backups:

Column Name	Description
VPS ID	The ID of the VPS whose backup has been found.
Source Node	The Node where the VPS was hosted during its backing up.
Creation date	The date and time when the backup was created.
Type	The backup type. An incremental backup represents only those changes that the VPS has undergone since its last full or incremental backup. Therefore, an incremental backup is usually much less in size than a full backup.
Backup Node	The Backup Node - the Node where the backup has been found.

Double-clicking on a VPS backup in this table brings about the **Virtual Private Server Backups** dialog where you can view extensive information about the current VPS backup, including all its full and incremental backups, as well as delete any of these backups or restore them in the manner depicted above.

Reinstalling Virtual Private Server

Reinstalling a Virtual Private Server is used if a VPS administrator has inadvertently modified, replaced, or deleted any file that is part of an application or OS template, which has brought about the VPS malfunction. You can reinstall the VPS in the two following ways:

- 1 The `vzctl recover` command restores the original VZFS symlinks of the VPS private area to the OS and/or application template(s) as they were at the time when the VPS was created and/or when the application template(s) were added to the VPS. This command does not deal with any user files on the VPS:

```
# vzctl recover 101
Recover OS template: redhat-9 20040116
```

- 2 The `vzctl reinstall` command creates a new private area for the problem VPS from scratch using its configuration files and its OS and application templates. Thus, a clean working copy of the VPS is created:

```
# vzctl reinstall 101
Optimizing VPS private area...
Calculating VPS disk usage...
Creating VPS private area
Getting private area from cache: /vz/private/pool/redhat-9/5
Calculating VPS disk usage...
Copying VPS credentials...
Old VPS file system was stored in /old
VPS re-installation completed successfully
```

Note: If any of the VPS application templates cannot be added to the VPS in a normal way, the reinstallation process will fail. This may happen, for example, if an application template was added to the VPS using the `--force` option of the `vzpkgadd` command (for more information on the `vzpkgadd` command see the `vzpkgadd` subsection of the [Virtuozzo Command Line Interface](#) chapter in the [Virtuozzo Reference Guide](#)).

In order to retain the personal data inside the old VPS, the utility also copies the contents of the old private area to the `/old` directory of the new private area (unless the `--skipbackup` option is given). The personal data can then be copied to the corresponding directories of the new private area and the `/old` directory eventually deleted:

```
# vzctl start 101
Starting VPS ...
VPS is mounted
Adding IP address(es): 10.0.186.101
Hostname for VPS set: ve101
File resolv.conf was modified
VPS start in progress...
# vzctl exec 101 ls /
bin
boot
dev
[...other directories...]
old
[...other directories...]
tmp
usr
var
```

Both the `vzctl recover` and `vzctl reinstall` commands retain the users' credentials base, unless the `--resetpwdb` option is specified.

Note: In the current version of Virtuozzo, VZMC does not support recovering VPSs; this functionality is accessible only thru the command line on the Hardware Node.

Customizing VPS Reinstallation

The default reinstallation, as performed by the `vzctl reinstall` command, creates a new private area for the broken VPS as if it were created by the `vzctl create` command and copies the private area of the broken VPS to the `/old` directory in the new private area so that no file is lost. There is also a possibility of deleting the old private area altogether without copying or mounting it inside the new private area, which is done by means of the `--skipbackup` option. This way of reinstalling corrupted VPSs might in certain cases not correspond exactly to your particular needs. It happens when you are accustomed to creating new VPSs in some other way than just using the `vzctl create` command. For example, you may install additional software licenses into new VPSs, or anything else. In this case you would naturally like to perform reinstallation in such a way so that the broken VPS is reverted to its original state as determined by you, and not by the default behavior of the `vzctl create` command.

To customize reinstallation, you should write your own scripts determining what should be done with the VPS when it is being reinstalled, and what should be configured inside the VPS after it has been reinstalled. These scripts should be named `vps.reinstall` and `vps.configure`, respectively, and should be located in the `/etc/sysconfig/vz-scripts` directory on the Hardware Node. To facilitate your task of creating customized scripts, Virtuozzo is shipped with sample scripts that you may use as the basis of your own scripts.

When the `vzctl reinstall vps_id` command is called, it searches for the `vps.reinstall` and `vps.configure` scripts and launches them consecutively. When the `vps.reinstall` script is launched, the following parameters are passed to it:

<code>--veid</code>	The ID of the VPS.
<code>--ve_private_tmp</code>	The path to the VPS temporary private area. This path designates where a new private area is temporarily created for the VPS. If the script runs successfully, this private area is mounted to the path of the original private area after the script has finished.
<code>--ve_private</code>	The path to the VPS original private area.

You may use these parameters within your `vps.reinstall` script.

If the `vps.reinstall` script finishes successfully, the VPS is started, and the `vps.configure` script is called. At this moment the old private area is mounted to the `/old` directory inside the new one irrespective of the `--skipbackup` option. This is done in order to let you use the necessary files from the old private area in your script, which is to be run inside the running VPS. For example, you might want to copy some files from there to regular VPS directories.

After the `vps.configure` script finishes, the old private area is either dismounted and deleted or remains mounted depending on whether the `--skipbackup` option was provided.

If you do not want to run these reinstallation scripts and want to stick to the default `vzctl reinstall` behavior, you may do either of the following:

- 1 Remove the `vps.reinstall` and `vps.configure` scripts from the `/etc/sysconfig/vz-scripts` directory, or at least rename them;

2 Modify the last line of the `vps.reinstall` script so that it would read

```
exit 128
```

instead of

```
exit 0
```

The 128 exit code tells the utility not to run the scripts and to reinstall the VPS with the default behavior.

Deleting Virtual Private Server

You can delete a Virtual Private Server that is not needed anymore with the `vzctl destroy <VPS_ID>` command. This command removes the Virtual Private Server private area completely and renames the VPS configuration file and action scripts by appending the `.destroyed` suffix to them.

A running VPS cannot be destroyed with the `vzctl destroy` command. The example below illustrates destroying VPS 101:

```
# vzctl destroy 101
Destroying VPS private area: /vz/private/101
VE is currently mounted (umount first)
# vzctl stop 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
# vzctl destroy 101
Destroying VPS private area: /vz/private/101
VPS private area was destroyed
# ls /etc/sysconfig/vz-scripts/101.*
/etc/sysconfig/vz-scripts/101.conf.destroyed
/etc/sysconfig/vz-scripts/101.mount.destroyed
/etc/sysconfig/vz-scripts/101.umount.destroyed
# vzctl status 101
VPSID 101 deleted unmounted down
```

If you do not need the backup copy of the VPS configuration files (with the `.destroyed` suffix), you may delete them manually.

Virtual Private Servers can be deleted by using VZMC. VZMC allows you to delete Virtual Private Servers that are not needed anymore. The VPSs that can be deleted are marked with grey monitor icons (user VPSs). To delete one or more user Virtual Private Servers, select it (them) in the **Virtual Private Servers** table in the right pane of the VZMC main window. You can use **CTRL+Click** to select or deselect an entry, **SHIFT+Click** to select a range of Virtual Private Servers, **CTRL+A** to select all Virtual Private Servers. Then right-click the selected VPS(s) and choose **Delete**. You can also press the **Delete** button on the toolbar or select **Delete** on the **Action** menu:

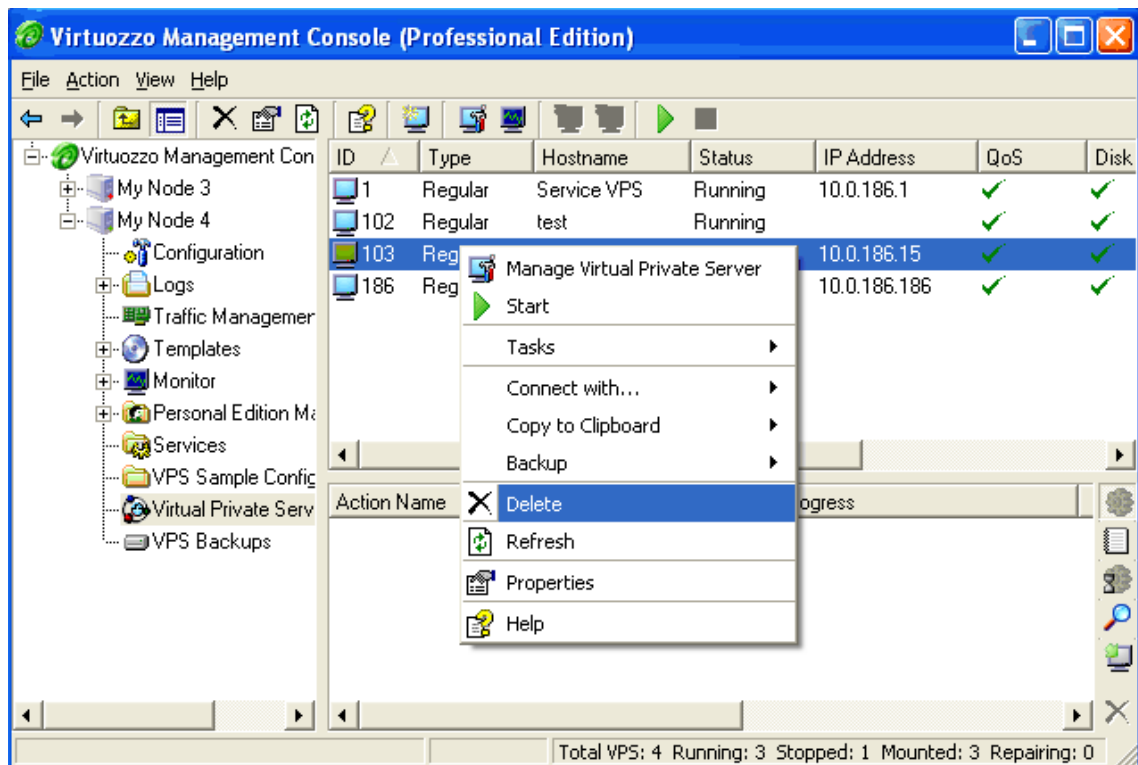


Figure 16: VZMC - Deleting VPS

Deleting a considerable number of Virtual Private Servers may take a rather long run. The progress is displayed in the **Actions** pane.

Disabling VPS

There may appear situations when you wish to forbid VPS owners to use their Virtual Private Servers. For example, it may happen in case the VPS owner uses it for unallowed purposes: intruding into computers of other users, participating in DoS attacks, etc.

In such cases, Virtuozzo allows you to disable a Virtual Private Server, thus, making it impossible to start the VPS once it was stopped. For example, you can execute the following command to disable VPS 101 residing on your Hardware Node:

```
# vzctl set 101 --disable yes
```

After the VPS stopping, the VPS user will not be able to start it again until you enable this VPS again by passing the `--disable no` option to `vzctl set`. You can also use the `--force` option to start any disabled Virtual Private Server. For example:

```
# vzctl start 101
VPS start disabled
# vzctl start 101 --force
Starting VPS...
VPS is mounted
Adding port redirection to VPS(1): 4643 8443
Adding IP address(es): 10.144.144.101
Hostname for VPS set: vps_101
VPS start in progress...
```

You can also enable/disable a Virtual Private Server by means of VZMC. To this effect, you should select the **Virtual Private Servers** item under the Hardware Node name on the VZMC main menu, right-click the corresponding VPS, and choose **Tasks --> Enable/Disable** on the context menu, respectively:

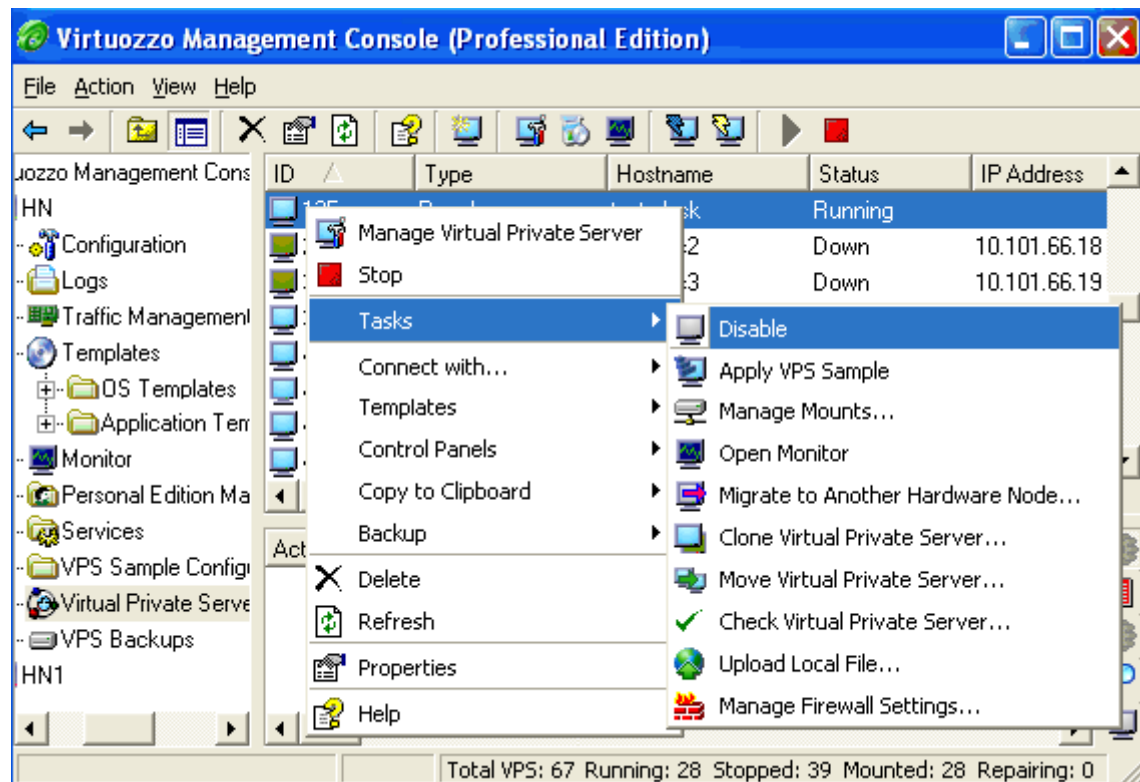


Figure 17: VZMC - Enabling/Disabling VPS

You can use CTRL+Click to select or deselect an entry, SHIFT+Click to select a range of Virtual Private Servers, CTRL+A to select all Virtual Private Servers.

Running Commands in Virtual Private Server

Usually, a Virtual Private Server administrator logs in to the VPS via network and executes any commands in the VPS as on any other Linux box. However, you might need to execute commands inside Virtual Private Servers bypassing the normal login sequence. This can happen if:

- You do not know the Virtual Private Server login information, and you need to run some diagnosis commands inside the VPS in order to verify that it is operational.
- Network access is absent for a Virtual Private Server. For example, the VPS administrator might have accidentally applied incorrect firewalling rules or stopped SSH daemon.

Virtuozzo allows you to execute commands in a Virtual Private Server in these cases. Use the `vzctl exec <VPS_ID>` command for running a command inside the VPS with the given ID. The session below illustrates the situation when SSH daemon is not started:

```
# vzctl exec 101 /etc/init.d/sshd status
sshd is stopped
# vzctl exec 101 /etc/init.d/sshd start
Starting sshd:[ OK ]
# vzctl exec 101 /etc/init.d/sshd status
sshd (pid 26187) is running...
```

Now VPS users can log in to the VPS via SSH.

When executing commands inside a Virtual Private Server from shell scripts, use the `vzctl exec2` command. It has the same syntax as `vzctl exec` but returns the exit code of the command being executed instead of the exit code of `vzctl` itself. You can check the exit code to find out whether the command has completed successfully.

If you wish to execute a command in all running VPSs, you can use the following script:

```
# for i in `cat /proc/vz/veinfo | awk '{print $1}' | egrep -v '^0$'`; \
do echo "VE $i"; vzctl exec $i <command>; done
```

where `<command>` is the command to be executed in all the running VPSs. For example:

```
# for i in `cat /proc/vz/veinfo | awk '{print $1}' | egrep -v '^0$'`; \
do echo "VE $i"; vzctl exec $i uptime; done
VE 1
 2:26pm up 6 days, 1:28, 0 users, load average: 0.00, 0.00, 0.00
VE 101
 2:26pm up 6 days, 1:39, 0 users, load average: 0.00, 0.00, 0.00
[The rest of the output is skipped...]
```

CHAPTER 4

Managing Templates

Template (or package set) is a set of original application files, repackaged for mounting over Virtuozzo File System. There are two types of templates in Virtuozzo: operating system templates and application templates. *OS templates* are used to create new Virtual Private Servers. Typical examples of OS templates are templates containing Red Hat, SuSE, or any other Linux distribution. An OS template cannot be installed on a VPS, because any VPS is already based on some OS template (just as Debian Linux cannot be installed on Red Hat Linux). *Application templates* are installed on VPSs after the latter have been created; more than one application template may be installed on one and the same VPS. Examples of application templates are MySQL template, ProFTPd FTP server template, or a template with any other application (possibly more than one application in a single template) not included into the OS template for some reason.

Also, both kinds of templates may be created for performing updates. A *template update* allows you to add, remove, or update any package the template consists of. Any update may be selectively applied to Virtual Private Servers, i.e. VPSs may use the original, latest, or any intermediate version of the template independently. A typical example of a template update is a security update for an application (like openssh, sendmail, and so on).

Note: Virtuozzo standard OS and application templates have evolved to new OS and application EZ templates in Virtuozzo 3.0. Although, the current version of Virtuozzo does support both standard and EZ templates, the latter demonstrate much more flexibility and manageability than their counterpart and represent the preferred way to manage templates on the Hardware Node and inside your Virtual Private Servers. Detailed information on OS and application EZ templates is provided in the **Managing EZ Templates** chapter (on page 93).

In This Chapter

Template Lifecycle	76
Listing Templates.....	77
Creating and Installing Application Templates.....	78
Creating and Installing Template Updates.....	79
Adding Templates and Template Updates to VPSs	81
Removing Templates and Template Updates From VPSs	82
OS Template Updates vs Application Templates	83
Creating OS Template With vzsnapshot	84
Creating OS Template.....	85

Template Lifecycle

Any template of any kind should be *created* first. The `vzpkgcreat` utility allows you to create a template. In addition, a number of already created templates are shipped by SWsoft with Virtuozzo.

Then, the template should be *installed on the Hardware Node* using the `rpm -i` command. For OS templates, as well as for OS template updates, the `vzpkgcache` program should be launched to *cache the new template into a tarball* for a precipitated creation of new VPSs based on this template.

The template that has been installed on the Hardware Node, may be either *added to any number of VPSs* with `vzpkgadd` (for application templates or their updates), or *a new VPS may be created* on the basis of an OS template with the `vzctl create` command.

Any template excluding OS templates may be *removed from the VPS* with the `vzpkgrm` command.

Finally, a template that is not used by any Virtual Private Server may be completely removed from the Hardware Node with the `rpm -e` command.

All these steps may be performed either thru the Virtuozzo command-line utilities indicated above, or with the help of VZMC.

Listing Templates

The `vzpkgls` utility allows you to list the templates installed on the Hardware Node. They may be already used or not used by certain VPSs:

```
# vzpkgls
redhat-9      20040116
proftpd-rh9  20040119
```

As you see, the `redhat-9` and `proftpd-rh9` templates are available on the Hardware Node. Specifying a VPS number as the parameter, this command prints the templates used by the specified VPS:

```
# vzpkgls 101
redhat-9 20040116
```

In Virtuozzo Management Console, it is sufficient to click on the friendly name of your Hardware Node in the left pane, and choose consecutively the **Templates** --> **Standard Templates** --> **Application Templates** or **OS Templates** options to see a list of the application or OS templates installed on the Node. For example:

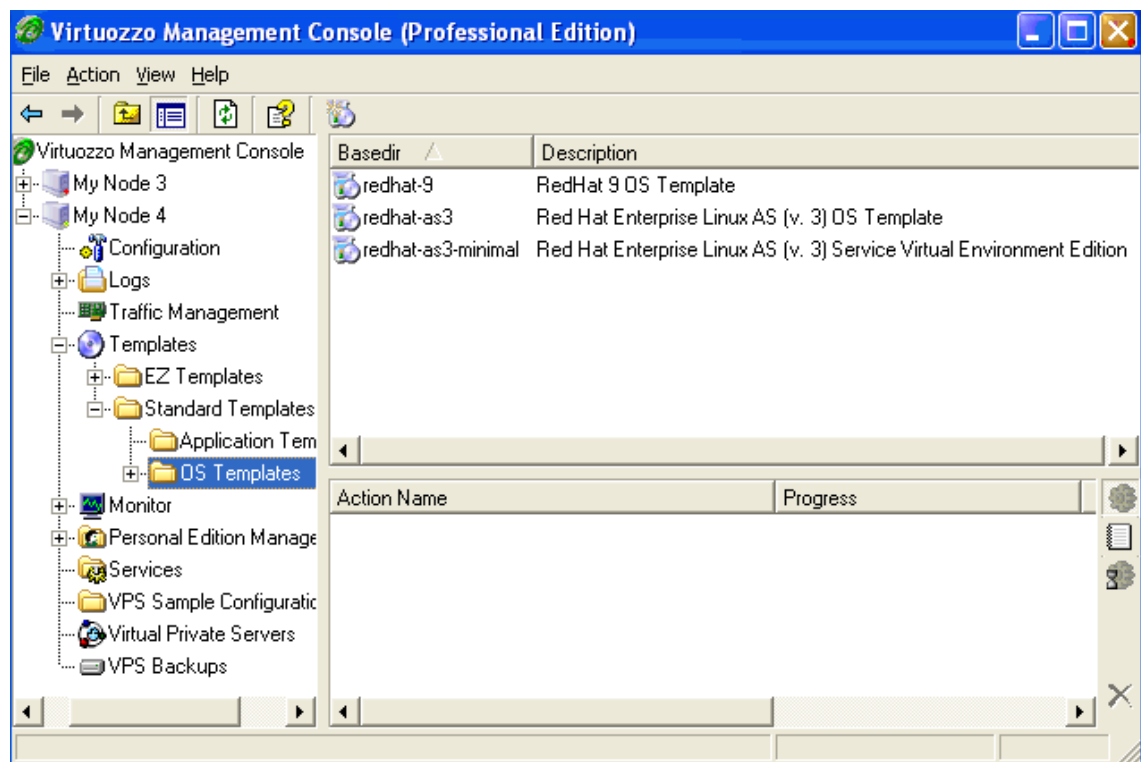


Figure 18: VZMC - Listing Templates

To see the templates used by a particular VPS, double-click the needed Virtual Private Server in the VPS list in the right pane to open this VPS management window, and then again choose **Templates** --> **Standard Templates** --> **Application Templates** or **OS Templates** in the left pane of the new window.

Creating and Installing Application Templates

As you may notice, the `samba` server and client applications are not included into the `redhat-7.2` template. However, if any Virtual Private Server needs this application, a new application template may be created.

The `vzpkgcreat` utility serves to create any template. In the example below, the `samba` RPM files from `redhat-7.2` are used to create a new application template compatible with the `redhat-7.2` OS template. The template created is the *base* template (unlike a template update); this is specified with the `-b` command line option. The `-s` option specifies the OS template the application template being created is compatible with. The `-n` and `-d` options specify the template name and description, correspondingly.

To create this template, you will need Red Hat Linux 7.2 CD 1 mounted:

```
# mount /mnt/cdrom
# cd /mnt/cdrom/RedHat/RPMS/
# vzpkgcreat -b -s redhat-7.2 -n 'samba template'\
-d 'samba server and client template for Red Hat 7.2' samba\
samba-2.2.1a-4.i386.rpm samba-client-2.2.1a-4.i386.rpm\
samba-common-2.2.1a-4.i386.rpm -o /tmp
```

The new template is created in the directory specified by the `-o` option. Use the `rpm -ivh` command to install this template on the Hardware Node:

```
# rpm -ivh /tmp/samba-template-20020920-1.0-1.i386.rpm
Preparing... ##### [100%]
1:samba-template-##### [100%]
# vzpkgls
redhat-7.2 20020902
debian-3.0 20020809
samba-2.2.1a 20020920
```

As you see, the `samba-2.2.1a` template is now available for installation on any VPS(s).

VZMC provides a special wizard for creating Virtuozzo templates. The **Create Virtuozzo Template** wizard will guide you through the process, gather all the necessary information, and prompt about possible inconsistencies. In order to invoke the wizard, click the **Create Virtuozzo Application or OS Template** link on the Hardware Node summary page (alternatively, you can right-click on the **Templates** item in the main tree and select the **Create Virtuozzo Template** option on the context menu or select **Templates** in the main tree and use the **Create Virtuozzo Template** button at the top toolbar).

Using the VZMC wizard for creating an application template has certain specifics in the respect that you have to use not the Hardware Node itself but the computer on which VZMC is installed to provide the valid path to the RPM packages from which you are going to build the template. Thus, even if your VZMC runs under Microsoft® Windows®, you had better insert the CD-ROM with the required RPMs into the CD-ROM drive of that very computer; otherwise, you may have trouble providing a valid path, for example, to a Linux computer.

Creating and Installing Template Updates

Often, the system administrator needs to upgrade packages included in a template. This may happen, for example, in case the distribution vendor provides a security update. If such an update pertains to an application added from a template to certain Virtual Private Servers, these Virtual Private Servers should be updated to the new version with the corresponding bug fixed. The easiest way to do it is to create the corresponding template update, install it on the Hardware Node and add it to the needed Virtual Private Servers.

Note: Sometimes the auto-update feature of certain applications updates these applications directly inside the corresponding Virtual Private Servers, thus ignoring the file sharing capabilities of Virtuozzo. To restore file sharing in such cases, the `vzcache` utility should be used, as is described in the **Cleaning Up Virtual Private Servers** subsection (on page 124).

E.g., the `redhat-7.2` OS template contains Version 3.1p1-6 of the `openssh` package. In the example below, a template update with `openssh-3.4p1-1` is created and installed.

For this template, you will need the `openssh` binary RPM files compiled for `redhat-7.2`:

```
# vzpkgcreat -c redhat-7.2 \
  openssh-3.4p1-1.i386.rpm openssh-server-3.4p1-1.i386.rpm \
  openssh-clients-3.4p1-1.i386.rpm -o /tmp
[ output skipped ...]
# rpm -i /tmp/redhat-7.2-template-20020920-1.0-1.i386.rpm
# vzpkgls
redhat-7.2 20020902 20020920
debian-3.0 20020809
samba-2.2.1a 20020920
```

Specify the `-c` option to indicate that the new template will be an OS template, and omit the `-b` option to indicate that you are creating a template update.

As you see, a new version of the `redhat-7.2` template (named 20020920) is available now. After installing this template, do not forget to run the `vzpkgcache` utility to create a new VPS cached archive with the latest version of the template.

Note: By default, if the template version is not specified explicitly, the `vzctl create` command creates Virtual Private Servers using the latest available version of the indicated OS template. Hence, from now on, new Virtual Private Servers in the example above will be created using the 20020920 version of the `redhat-7.2` template.

Virtuozzo Management Console uses the same **Create Virtuozzo Template Wizard** for creating template updates. This wizard is explained in the previous section. The specifics of installing a template update with this wizard in comparison with installing a base template are the following:

- 1 In the **Select Template Type** window, you should set the first radio button to **Template update**.

- 2 In the Template Parameters window, you should fill the Base directory field with the name of the already existing base directory of the template for which you are creating an update. For example:

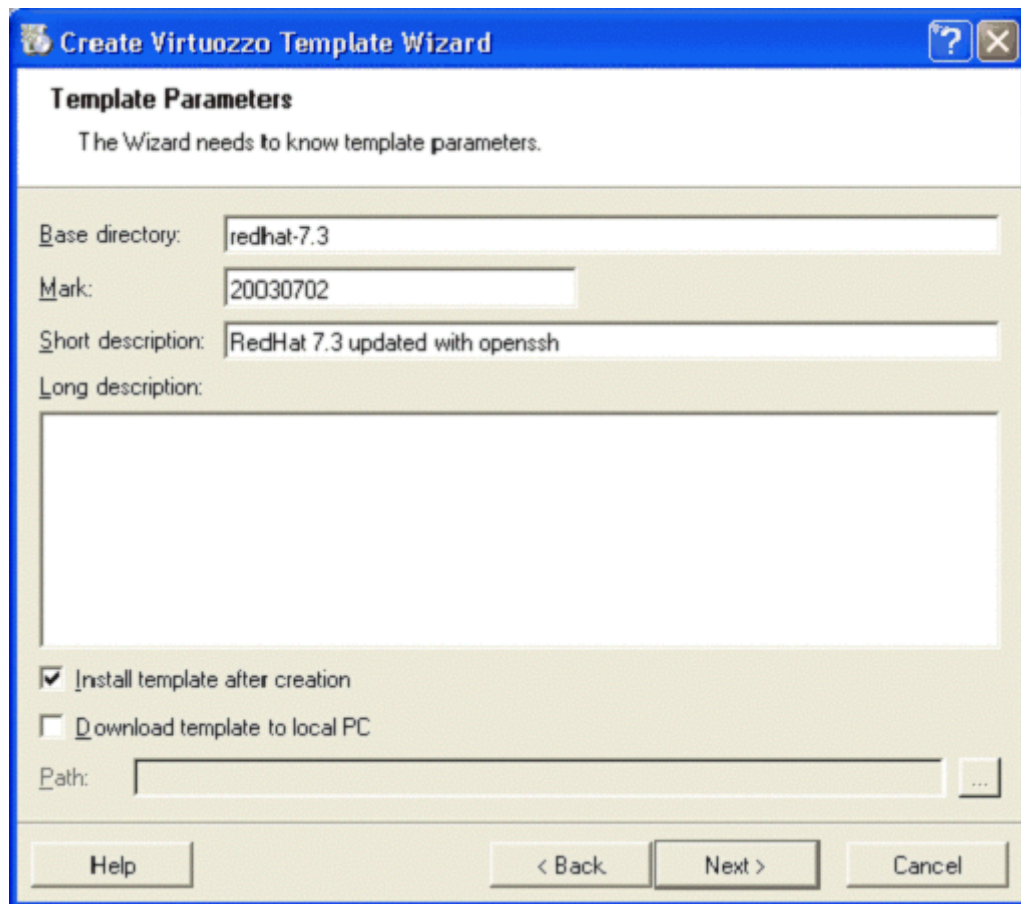


Figure 19: Creating VirtuoZZo Template Update - Specifying Base Directory

- 3 You will be suggested to specify not only those packages that are to be added to the template update, but also the packages to be removed from the base template.

In all other respects, using this wizard for creating base templates and template updates is similar.

Note: When creating and installing template updates, the packages dependencies are not checked. So, you should make sure beforehand that the packages you add to a template update do not have dependencies conflicts with the packages of the VPSs where the template update is to be added.

Adding Templates and Template Updates to VPSs

A number of application templates may be added to a VPS immediately upon the VPS creation. It happens if the sample configuration file on the basis of which the VPS is being created has the `TEMPLATES` parameter enumerating the necessary templates. To add application templates to an existing VPS, you should use the `vzpkgadd` utility.

The `vzpkgadd` utility handles the adding of both application templates and template updates to Virtual Private Servers. To successfully add a template to a VPS, this VPS should be running; otherwise, it is impossible to run the installation process in the VPS context.

In the example below, the `samba-2.2.1.a` application template is added to VPS 101:

```
# vzctl status 101
VEID 101 exist mounted running
# vzpkgadd 101 samba-2.2.1a
Updating VE 101 | ***** | [100%]
# vzpkgls 101
redhat-7.2 20020902
samba-2.2.1a 20020920
```

Template updates may be applied to Virtual Private Servers in the same way. In this case, specify the version of the template. For example:

```
# vzpkgadd 101 redhat-7.2 20020920
Updating VE 101 | ***** | [100%]
# vzpkgls 101
redhat-7.2 20020902 20020920
samba-2.2.1a 20020920
```

Now, VPS 101 uses the updated Red Hat 7.2 template, for example, with the `3.4p1-1` version of the `openssh` application, coming from the OS template update.

In VZMC, you can do it by performing the following operations:

- 1 Expand the **Templates** --> **Standard Applications** --> **Application Templates** item under the name of the Hardware Node where the needed templates are installed;
- 2 Select all the templates holding down the **CTRL** or **SHIFT** keys where necessary;
- 3 Right-click the selection and choose the **Install Into Virtual Private Servers** option;
- 4 Follow the instructions of the wizard.

If you are adding a template to only one VPS, you can as well do the following:

- 1 Open the list of Virtual Private Servers in the VZMC main window by selecting the **Virtual Private Servers** item in the Hardware Node tree;
- 2 Double-click the name of the VPS where you want to add a template;
- 3 Right-click on the **Templates** item in the main tree of the opened Virtual Private Server Manager, and select the **Add Virtuozzo Application Template** option on the context menu;
- 4 Follow the instructions of the wizard.

Notes: 1. If a template update has been created without manually checking its possible dependencies conflicts with the VPS software, adding this template to a VPS might fail. You should create a template update with compatible packages.

2. The Virtuozzo version for the 64-bit processors allows you to add only 64-bit application templates to your Virtual Private Servers.

Removing Templates and Template Updates From VPSs

The Virtuozzo `vzpkgm` utility allows either to downgrade a template added to a VPS or to completely remove it. Thus, if VPS users experience difficulties with `openssh-3.4p1-1`, it is possible to roll back this application to the original version (`openssh-3.1p1-6`). In this case, you should simply specify the destination version (mark) of the template:

```
# vzpkgm 101 redhat-7.2 20020920
[ output skipped ...]
# vzpkgls 101
redhat-7.2 20020902
samba-2.2.1a 20020920
```

By default, `vzpkgm` removes template updates only, performing rolling back to the previous template version. To remove a particular template from a VPS completely, the `-r` command line option should be specified. For example, after the `samba-2.2.1a` template installation, it can be removed from the Virtual Private Server in the following way:

```
# vzpkgm -r 101 samba-2.2.1a
Updating VE 101 | ***** | [100%]
# vzpkgls 101
redhat-7.2 20020902
```

Now, the original state of VPS 101 is restored – the `samba-2.2.1a` application template and `redhat-7.2` Version 20020920 template update have been removed from this VPS.

You can also remove templates by using VZMC. VZMC allows you to remove a template or template update from a Virtual Private Server. To this effect, you should:

- 1 Open a list of Virtual Private Servers in the VZMC main window by selecting the **Virtual Private Servers** item in the Hardware Node tree;
- 2 Double-click the name of the VPS where from you want to delete a template or a template update to open the Virtual Private Server Manager;
- 3 Click on the **Templates --> Standard Templates** item in the main tree of the opened Virtual Private Server Manager, and right-click on the template or template update you wish to delete in the right pane;
- 4 Select the **Delete** option on the context menu.

OS Template Updates vs Application Templates

When you want to provide additional packages to users inside several Virtual Private Servers, you can create either a new application template or an update of the OS template. Choosing between them, consider the following:

- When you create a new VPS, Virtuozzo installs an OS template with all updates by default, unless you manually specify the version you want.
- OS templates and their updates are cacheable, whereas application templates are not, which means that the deployment of an OS template may take less time.
- You should manually add separate application templates to Virtual Private Servers.
- If a number of Virtual Private Servers exist already on the Hardware Node, they will not be automatically updated with an update of the OS template. Either policy you choose, OS template updates or application templates, you need to add them manually to the existing Virtual Private Servers.

Essentially this means that if you want to keep your Virtual Private Servers up-to-date with the latest bug and security fixes from a vendor, then you will most likely want to create updates to the OS template. Thus, all new Virtual Private Servers based on this particular template will automatically benefit from the latest package versions. The vendor fixes for operating systems might also be quite large. For example, the size of updates for Red Hat 6.2 is approximately half the original distribution. If you package them as an additional template, the adding of such a template to a new VPS will take a considerable amount of time. And in most cases you do want all your new Virtual Private Servers to be patched with the latest packages.

Creating OS template updates is also a better option if OS templates shipped by SWsoft do not satisfy you completely. For example, if you want to have the French localization packages to be available in all your Virtual Private Servers, the easiest way of achieving it is to create an update for the OS template consisting of necessary RPMs.

Software packages that are unlikely to be installed in all Virtual Private Servers are better to be packaged as separate application templates. For example, application servers or commercial database software are usually packaged as application templates.

Creating OS Template With vzsnapshot

Using `vzsnapshot` is probably the easiest way of creating your first OS template. This utility is not installed by default during the Virtuozzo installation. If you are going to use this utility, install the `vzsnapshot` RPM located in the `HW/RPMS` subdirectory of Virtuozzo CD-ROM 2 with the `rpm -ihv` command to the Linux system of which you are going to take a snapshot. After that, just run `vzsnapshot <my_template_name>`. The utility will package the existing system installation as an OS template. By default, the template RPM will be placed in the `/VZSNAP` directory. You can override the default by setting the `VZSNAPSHOT` environment variable. The directory specified by `VZSNAPSHOT` should have twice as much free disk space as the used space on your system. Thus, if you have a small root partition, you had better set this variable.

The utility requires a significant amount of CPU time and disk space to produce the Virtuozzo template. It is recommended to decrease the user and process activity on the system whose snapshot is being taken: to stop unnecessary services, log out ordinary users and maybe even switch to the single user mode. The creation and/or deletion of files in the directories from which the template is being built can lead to a broken Virtuozzo template.

It is recommended to use `vzsnapshot` not inside any VPS, but in your root operating system. If you still wish to use the utility inside a VPS, make sure that

- the directory where the template will be placed is on a non-vzfs partition mounted into that VPS, and
- the `rawio` capability is set for that VPS (for details, see the **Configuring Capabilities** section of the **Advanced Tasks** chapter).

Notes:

You may create a symbolic link `/VZSNAP` pointing to the directory on the partition with sufficient disk space.

Taking a snapshot from a living system is a time-consuming process. It takes up to 1 hour depending on your system performance and the number of installed packages.

Using `vzsnapshot` is simple. However, the maintenance of an OS template created in this way is complicated – for example, the creation of template updates becomes much more difficult. Thus, in most cases, using templates prepared with `vzpkgcreat` (or obtained from SWsoft) is preferred. Please see the next section for information on using the standard Virtuozzo means of creating an OS template.

If you have upgraded to Virtuozzo 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

Creating OS Template

This section explains how to prepare an OS template from the packages obtained from a distribution vendor. Most Linux vendors allow to download the latest set of packages from their Web and FTP sites; alternatively, you can copy the packages from CD-ROMs or other media.

Creating an OS template can be divided into the following steps:

- Preparing a Virtual Private Server for building templates;
- Preparing the packages from a Linux distribution needed to create an OS template.
- Writing a template configuration file;
- Building and testing the template.

Preparing Virtual Private Server for Building OS Templates

It is not recommended to build OS templates on the Hardware Node. This will require installing a number of packages which are not needed for the normal operation of the Hardware Node and not required for running Virtual Private Servers. Therefore, it is recommended to assign a special Virtual Private Server for OS template creation.

Take into account the following considerations when creating a VPS for template building:

- It is recommended to use the corresponding template for the Virtual Private Server where you are going to build your OS template. For example, use the `redhat-9` template for the VPS where you are going to build a Red Hat 9 OS template; the `suse80` template for building a SuSE 8.0 OS template, and so on.
- The disk quota settings for the Virtual Private Server must be sufficient for distribution packages and resulting templates. Please note that all the packages from a single version of Red Hat Linux could occupy up to 2 Gb of disk space.
- After you have built a template, you might want to test the space its symlinks would occupy in a Virtual Private Server after the template is installed. The VZFS symlinks to the template files occupy much less space than the template files themselves, but they still occupy some space. However, you are not able to determine this space because these symlinks are seen as real files inside a VPS, and the space is reported as for the real template files. To see VZFS symlinks as symlinks, you have to bind mount a non-VZFS file system tree into the Virtual Private Server before installing the template. This is done with the `mount --bind` command, for example:

```
# mount --bind /tmp/tmp /vz/root/101/tmp
# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/hda3            20641788    6788924  12804224   35% /
/dev/hda1             1035660      42648    940404    5% /boot
none                 253364        0    253364    0% /dev/shm
/dev/hda2            13994990    1780463  11414490   14% /vz
/tmp/tmp             20641788    6788924  12804224   35% /vz/root/101/tmp
```

Now you may use the `ext3 /tmp` directory inside the VPS for working on the VZFS symlinks.

To prepare a Virtual Private Server for building OS templates, perform the following steps:

- 1 Create a new Virtual Private Server like in the example below:

```
# vzctl create 101 --pkgset redhat-9 --config vps.basic
Creating VE private area
[output of vzctl skipped...]
# vzctl set 101 --hostname ve101 --ipadd 10.0.186.101 --save
Saved parameters for VE 101
# vzctl set 101 --capability sys_rawio:on --save
Saved parameters for VE 101
```

- 2 Copy the `deb2vzdeb`, `dpkg`, `rpm`, `rpm-build`, `rpm2vzrpm`, and `tar` RPMs from Virtuozzo CD 1 and the `vzpkgtools` and `vzfsutil` RPMs from Virtuozzo CD 2 inside the Virtual Private Server.
- 3 Install the packages copied to the VPS with the `rpm -Uhv` command.

Preparing Packages

SWsoft provides quite a number of OS templates with Virtuozzo 3.0, all of which are thoroughly tested and include complete or near-complete functionality found in a standard OS distribution. As a rule, customized OS templates that you might wish to create will have some unneeded functionality curtailed, therefore, you should get rid of some packages from the original distribution before creating a template. However, excluding certain packages from the distribution is not the only thing you should do before deciding on the needed packages. There are several additional steps you should follow:

- 1 As far as there is no need in the `kernel`, `modutils`, and `mingetty` packages inside the VPS, they should be excluded, and dependencies on them should be resolved by building a `dummy.rpm` package providing them:

```
Provides: kernel = 2.4.20 modutils = 2.3.11-5 mingetty
```

The `dummy.rpm` post-install script should also have `modprobe` and `rmmod` calls commented out in the `iptables` `init` script.

- 2 Make sure that the OS packages you are planning to use include the following utilities: `sed`, `ed`, `bash`, `find`, `basename`, `grep`, `cat`. The latter are needed to successfully create VPSs on the basis of the would-be template and manage them with the help of `vzctl`.
- 3 Any package inside the VPS should be compatible with the corresponding package installed on the Hardware Node as to the base format and lock files. To avoid incompatibility, replace the eligible packages from your distribution with the packages of the same name found on Virtuozzo CD 1.
- 4 You may rebuild certain packages in order to exclude the functionality unnecessary in VPSs. For example, you may remove `acl` from `coreutils`, unnecessary devices from `dev`, `gpm` support from `ncurses`, etc.
- 5 Copy the resulting OS packages into the Virtual Private Server you prepared at the previous step, say, to the `/distro` directory. It is a good idea to check the list of the packages you plan to include in the template to have all dependencies resolved. The example below illustrates how this can be done:

```
[root@ve101 distro]# mkdir -p /tmp/tmp/var/lib/rpm
[root@ve101 distro]# rpm --initdb --root /tmp/tmp
[root@ve101 distro]# rpm --test -ihv `cat /tmp/list` --root /tmp/tmp
Preparing... ##### [100%]
```

This example creates an empty RPM database in `/tmp/tmp/var/lib/rpm` and verifies that the list of packages to be installed (in `/tmp/list`) has all dependencies resolved.

Writing Template Configuration File

The template configuration file defines the template contents and instructions on how the packages comprising the template shall be installed. It is an ASCII file consisting of several sections denoted by the `%` symbol at the beginning of the line. Empty lines and lines starting with `#` are ignored.

The sections and their descriptions are listed below:

Section	Description
name	Template name.

description	Template description.
cached	Set this section to “yes” for OS templates and their updates and to “no” for application templates. If this section is omitted, its value is treated as “no”.
package_manager	<p>This section specifies the commands used when the template is applied and revoked. The following variables can be defined in this section:</p> <p>PKGMAN — the package manager command line utility. The default is “rpm”.</p> <p>PM_INSTCMD — specifies the command used to install a new package. The default is “\$PKGMAN -ivh --ignoresize”.</p> <p>PM_UPGRCMD — specifies the command used to update or substitute a package. The default is “\$PKGMAN -Uvh --ignoresize --oldpackage”.</p> <p>PM_RMVCMD — specifies the command used to remove or uninstall a package. The default is “\$PKGMAN -e”.</p> <p>PM_FORCEOPT — the force option to be passed to the package manager. If omitted from the file, this variable is set to “--force”.</p> <p>PM_NODEPSOPT — the option to be passed to the package manager to ignore possible broken dependencies. If omitted from the file, this variable is set to “--nodeps”.</p> <p>PM_NOSCRIPSOPT — the option to be passed to the package manager causing it to bypass the execution of the installation and uninstallation scripts of the package. The default is “--noscripts”.</p>
basedir	This is the base directory for the location of the template files relative to the directory specified by the <code>TEMPLATE</code> variable in the Virtuozzo global configuration file (by default, <code>/vz/template</code>).
base_version	The list of packages included in the template in the form <code><name>-<version>-<release></code> . Packages are listed one per line.
compatible_sets	This section is applicable only to application templates. It lists OS templates to which this application template can be applied with the <code>vzpkgadd</code> command. If this section is absent, then <code>vzpkgadd</code> adds the template without compatibility check.
actions	<p>This section defines several variables that are used by the template management utilities. The most important is <code>RPMS_FIRST</code>. It lists the packages that should be installed without resolving dependencies and without executing scripts. An OS template may require such a list to “bootstrap” the basic system before installing the main set of packages. The scripts from this section will be executed by the shell interpreter.</p> <p>If the <code>HAVE_PREINSTALL</code> or <code>HAVE_POSTINSTALL</code> variables are set to “yes” in this section, then the <code>preinstall_actions</code> or <code>postinstall_actions</code> functions will be executed prior to or after the installation of packages, correspondingly.</p>

The sections having the default value can be omitted from the configuration file; in this case, their default values are used. Sections not listed above are ignored.

The description below guides you step by step through the process of creating a template configuration file for the Red Hat Linux 9 template. First, you have to choose a name for the file. The common convention in Virtuozzo is to name template configuration files after the `%basedir` section of the configuration file with the `.conf` suffix appended. For example, the base directory for your Red Hat 9 template could be `redhat-9`, and the configuration file name `redhat-9.conf`.

The first several sections are self-explanatory:

```
#
# This is Red Hat Linux 9 template configuration file
#

%name
My custom OS template

%description
This is a custom Red Hat Linux 9 OS template, containing only the
packages I need

%basedir
redhat-9

%cached
yes
```

Since the default package manager is RPM, you can completely omit the `%package_manager` section. Create the list of packages you've prepared at the previous step and fill in the `%base_version` section. The list of packages might look as follows:

```
%base_version 20030702
anacron-2.3-17
anonftp-4.0-9
[the rest of the list is skipped...]
```

The common Virtuozzo convention is to assign the date of the template creation as the `base_version` (for example, 20030702) in the template configuration file.

The `%actions` section contains the `RPMS_FIRST` list and pre- and post-install functions. The `RPMS_FIRST` list is needed since the installation of some packages might presuppose the execution of installation scripts dependent on some other packages which, in turn, require the former packages be already installed. In this case the correct installation of these packages to an empty directory tree will fail because of the failure of packages scripts. The packages from this list are installed twice: the first time without executing any packages scripts (such as pre- and post-install scripts and triggers for RPMs), and the second time – when the packages required by their installation scripts have been installed. It is a good idea to copy the list from a suitable template shipped with Virtuozzo.

The `preinstall_actions` and `postinstall_actions` shell functions are called just before and immediately after the packages installation. Both of them receive a single argument, which is the path to the installation directory. The contents of the first function depends on the package manager used by the distribution; in case of RPM-based templates, it verifies that the installation directory exists and that it is empty, and then creates the `/var/lib/rpm` subdirectory for the RPM database and the `/dev/null` device for redirecting the RPM output.

The `postinstall_actions` function can be customized as needed. The `postinstall_actions` function implemented in the `redhat-9` template shipped with VirtuoZZo does the following:

- Copies the Hardware Node time zone settings to the VPS;
- Copies the Hardware Node language settings to the VPS;
- Creates a link from `/etc/mtab` to `/proc/mounts`;
- Creates the shadow password database;
- Enables autostart for necessary services;
- Disables unnecessary services.
- Removes the `mingetty` and `bdflysh` entries from `/etc/inittab`;
- Changes the default Apache configuration to more reasonable settings;
- Disables certain resource-consuming scripts from `cron` (e.g. `updatedb`);
- Writes necessary information to `/etc/fstab`.

It is strongly recommended that you copy the `%actions` section from a suitable VirtuoZZo template and modify it as needed.

Building and Testing OS Template

Having the template configuration file ready, you can use the `vzpkgcreat` utility to create the template RPM. If the template configuration file contains no errors, building the template is straightforward:

```
[root@ve101 tmp]# vzpkgcreat -p /distro/ /tmp/redhat-9.conf
[most output skipped...]
Wrote: /usr/src/redhat/RPMS/i386/redhat-9-template- \
20030702-1.0-1.i386.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.36551
+ umask 022
+ cd /usr/src/redhat/BUILD
+ rm -f vzpkgfiles.lst
+ exit 0
```

RPM will place the template into the default directory (`/usr/src/redhat/RPMS/i386` on Red Hat Linux systems) unless you override it with the `-o` option of `vzpkgcreat`.

The easiest way to test the newly created OS template is to install it and run the `vzpkgcache` utility. This utility creates a Virtual Private Server based on this template and archives its private area content for speeding up consequent VPS creations based on this template. Remember that if you are testing your template inside a VPS, the `/vz` directory should reside on a non-VZFS file system and the `sys_rawio` capability should be granted to the VPS.

```
[root@ve101 tmp]# df
Filesystem            1k-blocks      Used Available Use% Mounted on
vzfs                  1048576       744817    303759   72% /
ext2                   8957295      2136253    6308971   26% /vz
[root@ve101 tmp]# rpm -ihv /usr/src/redhat/RPMS/i386/redhat-9- \
template-20030702-1.0-1.i386.rpm
Preparing...          ##### [100%]
 1:redhat-9-templat##### [100%]
[root@ve101 tmp]# vzpkgcache
Checking: version 20030702 of redhat-9.
Creating cache for redhat-9/20030702
Wrote cache for redhat-9/20030702
```

After deploying the template to the Hardware Node and running `vzpkgcache`, you can start creating Virtual Private Servers based on this template.

Creating OS Template in VZMC

In VZMC, the process of building OS templates does not differ from building application templates. You do not have to (though you may) create a template configuration file in advance; the **Create Virtuozone Template Wizard** can make it for you. Just launch the wizard and follow its instructions in the same way as is described in the **Creating and Installing Application Templates** section of this chapter. The only difference will be that you will not have to specify the OS templates with which the created template is to be compatible, which is obvious.

If you check the **Install template after creation** checkbox in the **Template Parameters** window, and the template is created and installed successfully on the **Hardware Node**, chances are that it is valid. To make sure of it and to prepare the template for creating Virtual Private Servers on its basis, you should cache it after the installation. To this effect, select **Templates** under the **Hardware Node** name in the left pane of the VZMC window, right-click on the **OS Templates** item, and select the **Cache OS Templates** option.

Note: If you have upgraded to Virtuozone 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

CHAPTER 5

Managing EZ Templates

The given chapter familiarizes you with EZ templates introduced in Virtuozzo 3.0 for the first time. Virtuozzo EZ templates have evolved on the basis of Virtuozzo standard templates and provide significant technology enhancements over their counterpart.

In This Chapter

Understanding EZ Templates.....	93
EZ Template Lifecycle.....	98
Preparing OS EZ Template for VPS Creation	98
Creating VPS on Basis of OS EZ Template.....	99
Installing Application EZ Templates on Hardware Node.....	103
Adding Application EZ Templates to VPS	104
Listing EZ Templates.....	107
Updating EZ Templates in VPS.....	108
Removing Application EZ Template From VPS	109

Understanding EZ Templates

Overview

Along with Virtuozzo OS and application templates as they are known in Virtuozzo 2.6.2 and earlier versions, Virtuozzo 3.0 introduces a new concept of EZ templates. In respect of their functionality and behaviour, EZ templates are identical to Virtuozzo standard templates, i.e. they also provide a way of sharing resources among lots of Virtual Private Servers by making use of Virtuozzo File System (VZFS) and thus enabling huge savings in terms of disk space and memory. However, EZ templates demonstrate a number of distinctive features as compared to their counterpart:

- EZ templates do not carry all the necessary package files inside themselves. They contain only information about what packages should be installed on the Hardware Node to make the templates fully operational and from what network repository these packages should be downloaded.

Note: For the sake of brevity, we will be saying throughout this guide that packages are included in EZ templates, which actually means that EZ templates contain the information on the corresponding packages without carrying the packages themselves.

- The dependencies of RPM packages included in an EZ template are automatically resolved during the packages installation on the Hardware Node. So, if the specified RPMs require other packages to be installed, these packages are also downloaded from the repository and installed on the Node. In case an RPM package has requirements that conflict with existing software on the Node or any dependencies for the package being installed cannot be satisfied, the RPM installation process fails without making any changes to the system.
- The EZ templates technology allows you to use the original OS vendor's packages and to receive the updated RPM packages from a central repository right after their release without having to wait until they are packaged into templates by SWsoft and delivered to you.

One of the basic concepts in the EZ template technology is the concept of 'repository' where RPM packages for the given EZ template are stored. A repository is a prepared directory or web site containing software packages and index files for Linux operating systems and/or any of their applications. An example of such a repository is the repository located at the <http://mirrors.usc.edu/pub/linux/distributions/fedora/> web site and storing RPM packages for the Fedora Core releases. Using repositories gives you the following advantages:

- RPM packages included in the given EZ template do not contain versions, but only names (e.g. `bash`, `crontabs`). So, you always update any RPM package included in the EZ template to its latest version available in the repository.
- As a result of the fact that a list of RPM packages does not provide their versions, EZ templates do not have versions either (e.g. `redhat-as4-x86`). Thus, you install any EZ template on the Hardware Node only once and, after that, use the installed template to update the RPM packages inside any VPS where it is applied.
- You can create several EZ templates for one and the same Linux operating system and/or application. Any EZ template you are provided with has the default RPM packages set included in it and is called the base EZ template. However, you can make your own EZ templates (the so-called non-base EZ templates) which may differ from the corresponding base template:
 - In the number of packages included in these EZ templates;
 - In the number and location of repositories to be used for these EZ templates;
 - In the number and kind of scripts to be executed on different EZ template lifecycle stages, etc.

Non-base EZ templates should have their own names and are created by appending a random identifier to the base EZ template name. For example, if you wish your VPS to run Red Hat Enterprise Linux 4 and to function as a Linux-based server only, you can create the `redhat-as4-x86-server` EZ template and include only those packages in it that are needed for performing main server tasks. So, you can specify packages to be used for setting up file and print sharing and exclude all the packages for graphical interfaces (GNOME and KDE).

Virtuozzo provides you with a `vzpkg` tool allowing you to automatically locate and obtain the correct RPM packages for your EZ templates from one or several package repositories. The packages are downloaded from the repository and installed on the Hardware Node in one of the following cases:

- When creating a cache for an OS EZ template;
- When adding an application EZ template or package to the first Virtual Private Server;
- When updating EZ templates or RPM packages inside a destination Virtual Private Server.

Note: Detailed information on how to manage RPM package repositories is provided in the **Setting Up Repository for EZ Template** section of the **Virtuozzo Installation Guide**.

EZ Template Directory Structure

All EZ templates and the RPM packages included in them and installed on the Hardware Node are located in the so-called template area the path to which is set as the value of the `TEMPLATE` variable in the `/etc/sysconfig/vz` file. By default, the `/vz/template` directory is used. The template area includes two main subdirectories:

- The `cache` subdirectory where the tar archive of the potential private area of a VPS based on the corresponding OS EZ template is stored. The tar archive is created during the OS EZ template caching. Please keep in mind that the OS EZ template should be obligatorily cached before you can start creating Virtual Private Servers on its basis.
- The template directory having the name of `<os_name>/<os_version>/<arch>/` where:
 - `<os_name>` denotes the name of the Linux distribution for which the OS EZ template is created (e.g. `redhat`, `centos`, `fedora-core`);
 - `<os_version>` is the version of the Linux distribution specified as `<os_name>` (e.g. 3 or 4), and
 - `<arch>` denotes the microprocessor architecture where the OS EZ template is to be run (`x86`, `x86-64`, `ia64`).

For example, after installing the 32-bit version of the Fedora Core 4 EZ template, the `/vz/template/fedora-core/4/x86/` directory on the Hardware Node is created.

In its turn, the `<arch>` directory contains the following subdirectories and areas:

- The template configuration subdirectory including:
 - the `config/os/default` directory where the meta data for the base OS EZ template are stored.
 - the `config/os/<setname>` directory where the meta data for non-base OS EZ templates, if any, are stored.
 - the `config/app/<app_name>/default` directory where the meta data for the base application EZ template are stored. This directory is created if at least one application EZ template for the given OS EZ template is installed on the Hardware Node.
 - the `config/app/<app_name>/<setname>` directory where the meta data for non-base application EZ templates, if any, are stored.

- The packages area containing a number of RPM packages downloaded from the repository and installed on the Hardware Node. The installed files can be shared among Virtual Private Servers, i.e. when a VPS based on the given OS EZ template is created or application EZ templates are added to any VPS, it contains only symlinks to the template files in the packages area. The installed RPM package has the following structure:

```
<name>-<epoch>:<version>-<release>.<arch>
```

where:

- *<name>* is the package name;
- *<epoch>* denotes the package epoch;
- *<version>* indicates the package version;
- *<release>* is the package release, and
- *<arch>* denotes the microprocessor architecture where the package is to be used.

Examples of the installed RPM package are the `flex-2.5.4a-33.i386` or `glib-2.3.5-10.i386` packages that can be found in the packages area on the Hardware Node after installing and caching the Fedora Core 4 OS EZ template.

- One or several subdirectories including the cached meta data and packages of the OS EZ template. The directories have the names of:
 - *baseN* for the base OS template;
 - *<setname>N* for the non-base OS template with the name of *<setname>*, if any;
 - *<appname>N* for the base application template, and
 - *<appname>-<setname>N* for the application template with the name of *<setname>*, if any.

The *N* denotes the index number of the URL specified in the `repositories/mirrorlist` file (please see information on the `repositories/mirrorlist` file below).

As has been mentioned above, the configuration directory (i.e. `/<template_area>/<template_directory>/config`) contains a number of subdirectories storing the meta data for EZ templates. The contents of the subdirectories can vary depending on whether it is a base EZ template or a non-base one and on the EZ template type (OS or application template). The general contents of EZ templates can be described as follows:

- Data files:
 - `packages`: contains a list of RPM packages names included in the corresponding EZ template.
 - `package_manager`: specifies the packaging system used to handle the EZ template.
 - `repositories`: a list of repositories where the packages comprising the EZ template are stored.
 - `mirrorlist`: one or several URLs to the file containing a list of repositories from where the packages comprising the EZ template are to be downloaded.
 - `distribution`: the name of the Linux distribution for which the EZ template is created. This file should be absent for application EZ templates.
 - `summary`: brief information on the EZ template.

- **description:** detailed information on the EZ template. As distinct from the summary file, it can contain non-base data on the EZ template.
- **environment:** a list of environment variables set in the form of *key=value*.
- **Scripts:**
 - **pre-cache:** this script is executed before installing the RPM packages included in the EZ template on the Hardware Node.
 - **post-cache:** this script is executed after installing the RPM packages included in the EZ template on the Hardware Node.
 - **pre-install:** this script is executed before adding the EZ template to or installing the RPM package inside the Virtual Private Server.
 - **post-install:** this script is executed after adding the EZ template to or installing the RPM package inside the Virtual Private Server.
 - **pre-upgrade:** this script is executed before updating the RPM packages inside the Virtual Private Server.
 - **post-upgrade:** this script is executed updating the RPM packages inside the Virtual Private Server.
 - **pre-remove:** this script is executed before removing the application EZ template/RPM package from the Virtual Private Server.
 - **post-remove:** this script is executed after removing the application EZ template/RPM package from the Virtual Private Server.
- **Document files:** one or several files with arbitrary names containing the information on the EZ template.

Note: Detailed information on the files contained in the OS template configuration directory is provided in the EZ Template Management Utilities chapter of the Virtuozzo Reference Guide.

While working with the files in the configuration directory, please keep in mind the following:

- The `packages` file should be specified for all EZ templates.
- The `packages`, `package_manager`, and `repositories/mirrorlist` files should be specified for all base OS EZ templates.
- The `package_manager` and `distribution` files should be specified for all base OS EZ templates and absent for non-base OS EZ templates and all application EZ templates.
- The information from the `repositories/mirrorlist` file(s) created for non-base OS and all application EZ templates is added to that in the `repositories/mirrorlist` file(s) for the base OS EZ template.
- If there is no script for non-base OS and all application templates, the corresponding script for the base OS EZ template will be executed.

EZ Template Lifecycle

An EZ template has the following development stages:

- 1 Any EZ template should be first installed on the Hardware Node. The `rpm -i` command enables you to install OS and application EZ templates on the Node.
- 2 The EZ template should be cached by using the `vzpkg create cache` command. This step is required for OS EZ templates only. As a result of the `vzpkg create cache` execution, the necessary packages included in the OS EZ template are downloaded from the network repository, installed in the `/vz/template/<os_name>/<os_version>` directory on the Hardware Node, and a gzipped tarball for the OS EZ template is created and put to the `/vz/template/cache/` directory.
- 3 Either a new VPS is created on the basis of the cached OS EZ template with the `vzctl create` command or an application EZ template is added to any number of Virtual Private Servers by using the `vzpkg install` command. In the latter case, the necessary application files are first downloaded from the network repository and installed on the Hardware Node and then the VZFS links to the installed files are created and added to the VPS private area.
- 4 An obsolete EZ template applied to the corresponding VPS can be updated by using the `vzpkg update` command.
- 5 Any EZ template excluding OS EZ templates can be removed from the VPS with the `vzpkg remove` command.

All these steps may be performed either thru the Virtuozzo command line utilities indicated above or with the help of VZMC.

Preparing OS EZ Template for VPS Creation

OS EZ templates are used to create Virtual Private Servers on their basis. To prepare an OS EZ template for the VPS creation, you should:

- 1 Install the OS EZ template on the Hardware Node and
- 2 Cache the installed OS EZ template.

Note: If you have upgraded to Virtuozzo 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

To install a new OS EZ template on the Hardware Node, you should use the `rpm -i` command. For example, to install the Red Hat Enterprise Linux 4 OS EZ template on the Node:

```
# rpm -ihv redhat-as4-x86-tmpl-3.0.0-1.swsoft.noarch.rpm
Preparing...                               ##### [100%]
 1:redhat-as4-x86-tmpl                     ##### [100%]
# vzpkg list
redhat-as4-x86
```

As you see, the `redhat-as4-x86` EZ template is now installed on the Hardware Node. The corresponding path is `/vz/template/redhat/as4/`. However, before the `redhat-as4-x86` EZ template can be used as a basis for the VPS creation, it should be cached. This can be done by using the `vzpkg create cache` command:

```
# vzpkg create cache redhat-as-x86
...
Complete!
Packing cache file redhat-as4-x86.tar.gz ...
Cache file redhat-as4-x86.tar.gz [14M] created.
# ls /vz/template/cache
redhat-as4-x86.tar.gz
```

Note: Before you can start caching your OS EZ templates, you may need to set up a package repository for them. So, you have to build a special repository for all commercial versions of the Linux distributions (e.g. Red Hat Linux Enterprise 4). Detailed information on how to manage package repositories is provided in the [Setting Up Repository for EZ Template](#) chapter of the [Virtuozzo Installation Guide](#).

After the `redhat-as4-x86` EZ template has been successfully cached, you can start creating Virtual Private Server on its basis. Detailed information on how to create Virtual Private Servers on the basis of OS EZ templates is provided in the [Creating VPS on Basis of OS EZ Template](#) subsection (on page 37).

Creating VPS on Basis of OS EZ Template

The given section provides information on how to create Virtual Private Servers on the basis of OS EZ templates.

Choosing OS EZ Template

Before starting to create a Virtual Private Server, you shall decide on which OS EZ template your VPS will be based on. There might be several OS EZ templates installed on the Hardware Node and prepared for the VPS creation; use the `vzpkg list` command to find out what OS EZ templates are available on your system:

```
# vzpkg list -O
redhat-as4-x86          2005-11-21 23:59:44
fedora-core-4-x86      2005-12-20 12:45:52
```

The `-O` option passed to the `vzpkg list` command allows you to list only OS EZ templates installed on the Hardware Node. As you can see, the `redhat-as4-x86` and `fedora-core-4-x86` OS EZ templates are currently available on the Node. The time displayed beyond OS EZ templates indicates when the corresponding EZ template was cached.

You can also use the `--with-summary` option to display brief information on the installed OS EZ templates:

```
# vzpkg list -O --with-summary
redhat-as4-x86          Red Hat Enterprise Linux AS v. 4 OS template
fedora-core-4-x86      Fedora Core 4 OS template
```

For complete information on the `vzpkg list` command, you can consult the [Virtuozzo Reference Guide](#).

In VZMC, you only have to click consecutively the name of your Node, the **Templates** item, the **EZ Templates** item, and the **OS Templates** item to see a list of the installed OS EZ templates:

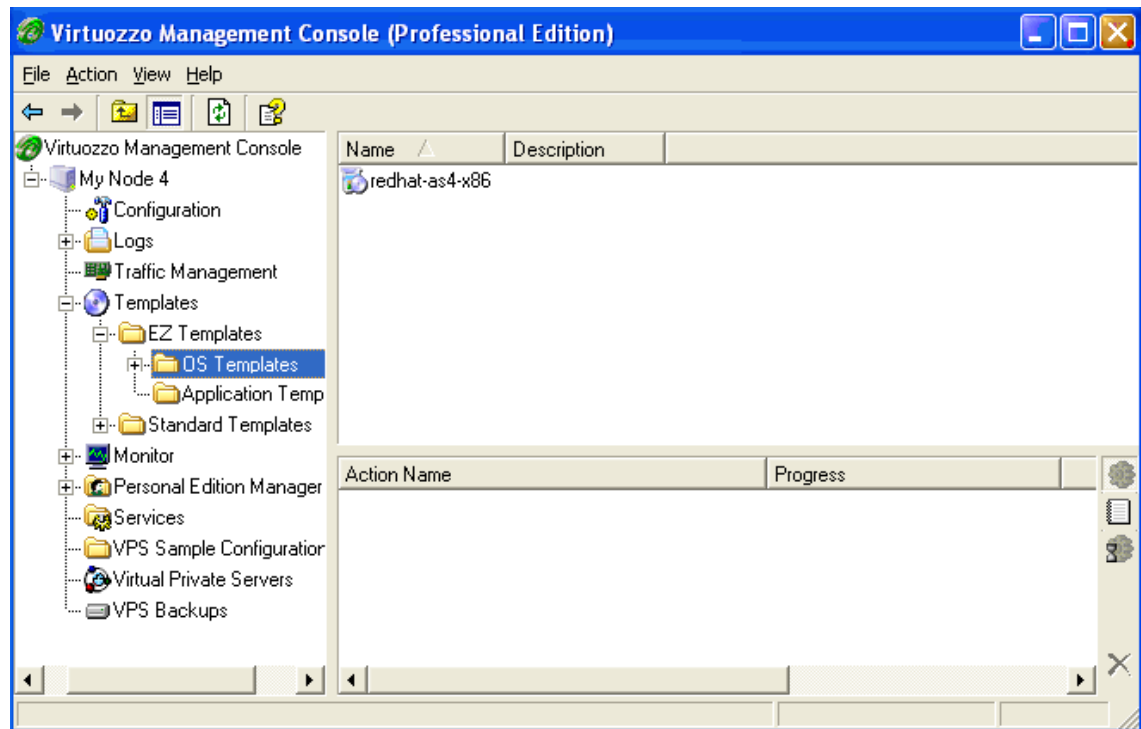


Figure 20: VZMC - Listing EZ OS templates

Note: If you have upgraded to Virtuozzo 3.0 with Linux kernel 2.4, you will not be able to use the templates of those OSs that are built for Linux kernel 2.6 only (Fedora Core 3 and 4, Red Hat Enterprise Linux 4, CentOS 4, etc.).

Creating Virtual Private Server

After the VPS ID and the installed OS EZ template have been chosen, you can create the VPS private area with the `vzctl create` command. The private area is the directory containing the VZFS symlinks, copy-on-write area, and private files of the given VPS. The private area is mounted to the `/vz/root/VPS_ID` directory on the Hardware Node and provides VPS users with a complete Linux file system tree.

The `vzctl create` command requires only the VPS ID and the name of the OS template as arguments; however, in order to avoid setting all the VPS resource control parameters after creating the private area, you can specify a sample configuration to be used for your new Virtual Private Server. The sample configuration files are residing in the `/etc/sysconfig/vz-scripts` directory and have names with the following mask: `vps-<configname>.conf-sample`. The most commonly used sample is the `ve-vps.basic.conf-sample` file; this sample file has resource control parameters suitable for most Virtual Private Servers.

Thus, for example, you can create a new VPS by typing the following string:

```
# vzctl create 101 --ostemplate redhat-as4-x86 --config vps.basic
Creating VPS private area (redhat-as4-x86)
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
Delete port redirection
Adding port redirection to VPS(1): 4643 8443
```

In this case, Virtuozzo will create a Virtual Private Server with ID 101, the private area based on the `redhat-as4-x86` OS EZ template, and configuration parameters taken from the `ve-vps.basic.conf-sample` sample configuration file.

If you specify neither an OS template nor a sample configuration, `vzctl` will try to take the corresponding values from the global Virtuozzo configuration file (`/etc/sysconfig/vz`). So you can set the default values in this file using your favorite text file editor, for example:

```
DEF_OSTEMPLATE=".redhat-as4-x86"
CONFIGFILE="vps.basic"
```

and do without specifying these parameters each time you create a new VPS. Please keep in mind that the `.` symbol before the template name in the `DEF_OSTEMPLATE` parameter is used to indicate that the VPS being created is to be based on an EZ, and not standard template.

Now you can create a VPS with ID 101 with the following command:

```
# vzctl create 101
Creating VPS private area (redhat-as4-x86)
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
Delete port redirection
Adding port redirection to VPS(1): 4643 8443
```

In principle, now you are ready to start your newly created Virtual Private Server. However, typically you need to set its network IP address, hostname, DNS server address and root password before starting the Virtual Private Server for the first time.

Installing Application EZ Templates on Hardware Node

The same way as you use an OS EZ template on the Virtuozzo system in order to create any number of Virtual Private Servers on its basis and share its resources, you can use application EZ templates in Virtuozzo in order to share package files among any number of VPSs. You can then add these applications to any number of Virtual Private Servers.

To install a new application EZ template on the Hardware Node, you should use the `rpm -i` command. For example, to install the `mysql` EZ template intended to be run on the Red Hat Enterprise Linux 4 distribution, you can issue the following command:

```
# rpm -ihv mysql-redhat-as4-x86-tmpl-3.0.0-2.swsoft.noarch.rpm
Preparing...                               ##### [100%]
 1:mysql-redhat-as4-x86                    ##### [100%]
```

The `mysql` EZ template will be installed in the `/vz/template/redhat/as4/x86/config/app/mysql` directory on the Hardware Node. To make sure that this EZ template has been successfully installed, you can use the `vzpkg list` command:

```
# vzpkg list
redhat-as4-x86                               2005-11-21 02:22:45
redhat-as4-x86  mysql
```

As you see, the `mysql` EZ template is now available on the Node and can be added to any number of Virtual Private Servers.

Adding Application EZ Templates to VPS

To add an application EZ template to an existing Virtual Private Server, you should use the `vzpkg install` command. To successfully add an application EZ template to a VPS, this VPS should be running; otherwise, it is impossible to run the installation process in the VPS context.

In the example below, the `mysql` application EZ template meant for the usage with Red Hat Enterprise Linux 4 and already installed on the Hardware Node is added to VPS 101:

```
# vzctl status 101
VPSID 101 exist mounted running
#vzpkg list
redhat-as4-x86
redhat-as4-x86    mysql
...
# vzpkg install 101 mysql
...
Installed:
mysql          i386          0:4.1.12-3.RHEL4.1
mysql-bench    i386          0:4.1.12-3.RHEL4.1
mysql-devel    i386          0:4.1.12-3.RHEL4.1
...
```

You can also add an application EZ template in VZMC by performing the following operations:

- 1 In the VZMC left pane, select **Templates --> EZ Templates --> Application Templates** under the corresponding Hardware Node name.

- 2 In the VZMC right pane, right-click the application EZ template you wish to add to your VPS and select **Install into Virtual Private Servers** on the context menu. The following window is displayed:

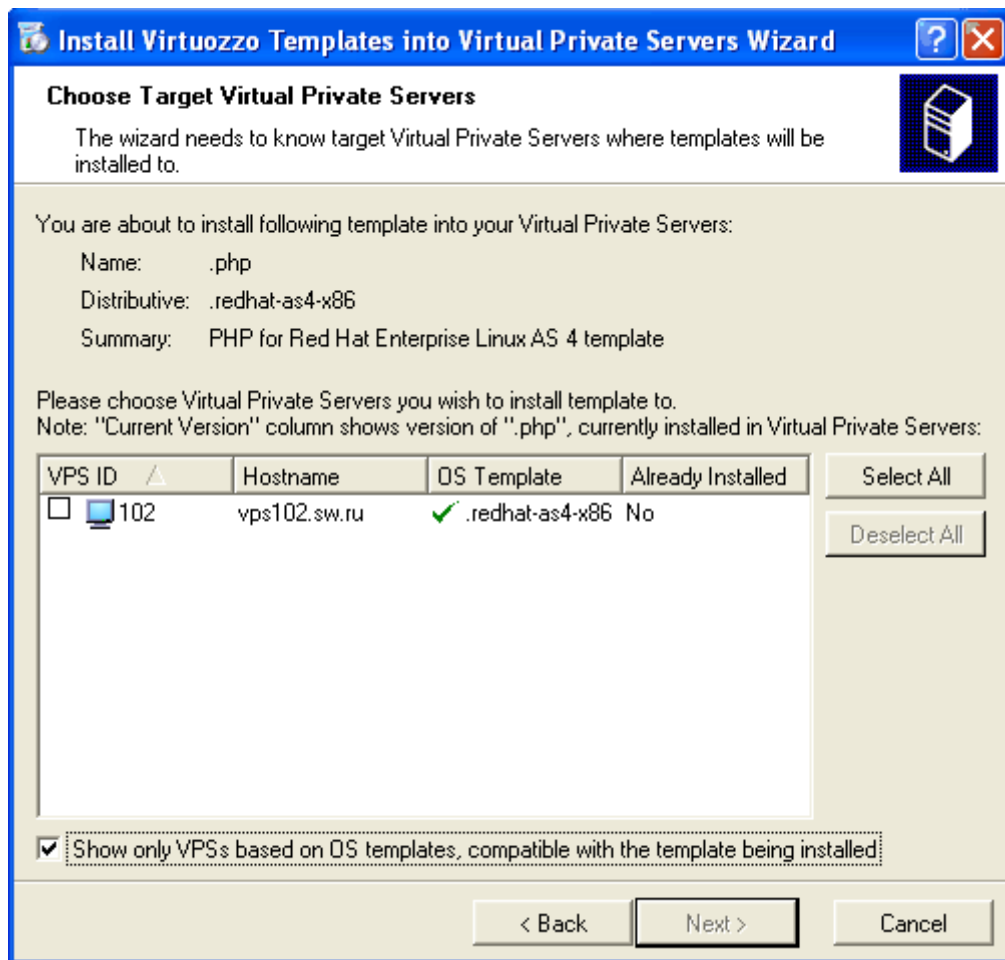


Figure 21: VZMC - Choosing VPS

- 3 Select the checkbox(es) of the VPS(s) where the application EZ template is to be added and click **Next**. You can use the **Select All/Deselect All** buttons to select/deselect all Virtual Private Servers listed in the **Choose Target Virtual Private Servers** window.

Note: To facilitate working with your application EZ templates, you can select the checkbox at the bottom of the window to display only the VPSs running the Linux distributions that are compatible with the given application EZ template.

- 4 On the next screen, you can choose the following options:
 - Select the **Force template installation** checkbox to force the EZ template installation inside the VPS. In this case no dependencies and no available versions of the application EZ template will be checked during its installation.
 - Select the **Use packages from local cache only** checkbox to use the packages needed for the EZ template installation from the local cache on the Hardware Node only. In this case you should make sure that all necessary files are present in your local directory; otherwise, the EZ template installation process will fail.

- 5 The last window allows you to review the information entered on the previous steps. If you are satisfied with the data entered, click **Finish** to start adding the application EZ template to the VPS; otherwise, click **Back** and change the necessary parameters.

If you are adding an EZ template to only one VPS, you can as well do the following:

- 1 Open a list of Virtual Private Servers in the VZMC main window by selecting the **Virtual Private Servers** item in the Hardware Node tree;
- 2 Double-click the name of the VPS where you want to add an EZ template;
- 3 Expand **Templates --> EZ Templates**, right-click the **Application Templates** item, and select the **Add Virtuozzo Application Template** option on the context menu;
- 4 Follow the instructions of the wizard.

Alternatively, you can open a list of VPSs in the VZMC main window, right-click the VPS where you wish to add an EZ template, select **Templates --> Add Virtuozzo Application Templates**, choose the needed EZ templates in the displayed window by using the **Add** button, and click **Install**.

Listing EZ Templates

The `vzpkg list` command allows you to list the EZ templates installed on the Hardware Node. They may be already used or not used by certain Virtual Private Servers:

```
# vzpkg list
redhat-as4-x86
fedora-core-4-x86          2005-12-20 14:15:14
```

As you see, the `redhat-as4-x86` and `fedora-core-4-x86` EZ templates are available on the Hardware Node. The characters opposite the `fedora-core-4-x86` EZ template informs you of the date and time when RPM packages included in the template were updated for the last time. In its turn, the characters absence beside the `redhat-as4-x86` EZ template indicates that the template has not yet been cached at all.

Specifying a VPS number as the parameter, `vzpkg list` prints the EZ templates used by the specified VPS:

```
# vzpkg list 101
fedora-core-4-x86          2005-12-20 14:15:14
```

In Virtuozzo Management Console, it is sufficient to click on the friendly name of your Hardware Node in the left pane and choose consecutively the **Templates --> EZ Templates --> OS Templates** or **Application Templates** options to see a list of the OS or application EZ templates installed on the Hardware Node, respectively. For example:

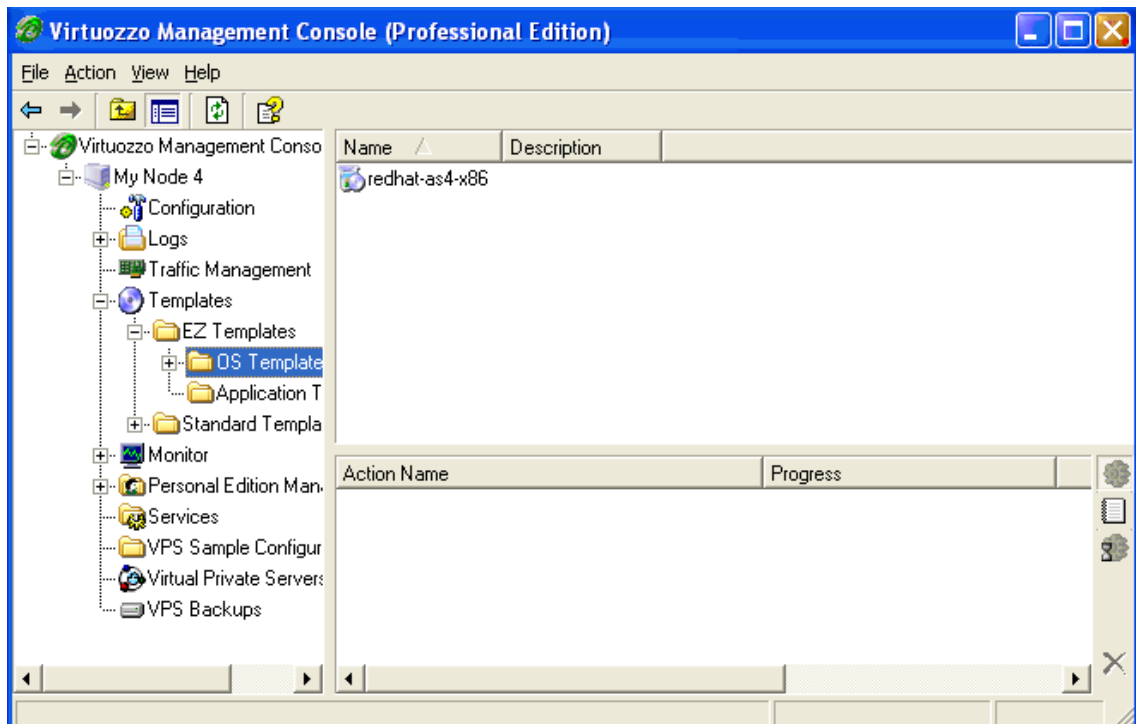


Figure 22: VZMC - Listing EZ Templates

To see the EZ templates used by a particular VPS, double-click the needed Virtual Private Server in the VPS list in the right pane to open this VPS management window, and then again choose **Templates --> EZ Templates --> OS Templates** or **Application Templates** in the left pane of the displayed window.

Updating EZ Templates in VPS

Virtuozzo 3.0 allows you to update the OS EZ template a Virtual Private Server is based on and any of the application EZ templates. You can do it by using the `vzpkg update` utility. So, you can issue the following command to update the `redhat-as4-x86` OS EZ template running inside VPS 101:

```
# vzpkg update 101 redhat-as4-x86
...
Updating: httpd ##### [1/4]
Updating: vzdev ##### [2/4]
Cleanup : vzdev ##### [3/4]
Cleanup : httpd ##### [4/4]

Updated: httpd.i386 0:2.0.54-10.2 vzdev.noarch 0:1.0-4.swsoft
Complete!
Updated:
httpd          i386          0:2.0.54-10.2
vzdev          noarch        0:1.0-4.swsoft
```

Note: A Virtual Private Server has to be running in order to update an EZ template inside this VPS.

As you can see from the example above, the `httpd` and `vzdev` applications have been updated for the `redhat-as4-x86` OS EZ template. If you wish to update all EZ templates (including the OS EZ template) inside Virtual Private Server 101 at once, you should execute the following command:

```
# vzpkg update 101
...
Running Transaction
  Updating : hwdata ##### [1/2]
  Cleanup  : hwdata ##### [2/2]

Updated: hwdata.noarch 0:1.0-3.swsoft
Complete!
Updated:
hwdata          noarch      0:0.158.1-1
```

In the example above, only the `hwdata` package inside VPS 101 was out of date and has been updated to the latest version.

In VZMC, you should perform the following operations to update the OS EZ template a Virtual Private Server is based on and/or any of its application EZ templates:

- 1 Open a list of Virtual Private Servers in the VZMC main window by selecting the **Virtual Private Servers** item in the **Hardware Node** tree.
- 2 Double-click the name of the VPS where you wish to add an EZ template to open the **Virtual Private Server Manager**.

- 3 Click on the **Templates --> EZ Templates** item in the main tree of the opened Virtual Private Server Manager.
- 4 In the right pane, double-click either the **OS Templates** or **Application Templates** folder depending on what EZ template you wish to update.
- 5 Right-click the corresponding EZ template and select the **Update Installed Packages** option on the context menu.
- 6 In the displayed window, select the checkboxes of the packages that are included in the EZ template and that you wish to update and click on the **Update** button.

Removing Application EZ Template From VPS

The Virtuozzo `vzpkg remove` command allows you to remove one or several application EZ templates from a Virtual Private Server. A session below shows you an example how to remove the `mysql` EZ template from VPS 101:

```
# vzpkg list 101
redhat-as4-x86                2005-11-21 02:21:56
redhat-as4-x86  mysql          2005-11-21 05:36:42
# vzpkg remove 101 mysql --with
vzpkg remove 101 mysql
Removed:
mysql
mysql-server
mysql-devel
mysql-bench
perl-DBD-MySQL
# vzpkg list 101
redhat-as4-x86                2005-11-21 02:21:56
```

You can see that the `mysql` EZ template has been successfully removed from VPS 101. The `--with` option tells the `vzpkg remove` command to also delete from the Virtual Private Server all packages having interdependencies with `mysql`.

To remove an application EZ template from a VPS in VZMC, do the following:

- 1 Open a list of Virtual Private Servers in the VZMC main window by selecting the **Virtual Private Servers** item in the Hardware Node tree;
- 2 Double-click the name of the VPS where from you want to delete a template to open the Virtual Private Server Manager;
- 3 Select **Templates --> EZ Templates --> Application Templates** and right-click on the EZ template to delete in the right pane;
- 4 Select the **Delete** option on the context menu.

CHAPTER 6

Managing Resources

The main goal of resource control in Virtuozzo is to provide Service Level Management or Quality of Service (QoS) for Virtual Private Servers. Correctly configured resource control settings prevent serious impacts resulting from the resource over-usage (accidental or malicious) of any Virtual Private Server on the other Virtual Private Servers. Using resource control parameters for Quality of Service management also allows to enforce fairness of resource usage among Virtual Private Servers and better service quality for preferred VPSs, if necessary.

In This Chapter

What are Resource Control Parameters?.....	110
Managing Disk Quotas.....	111
Managing CPU Share.....	129
Managing Network Accounting and Bandwidth.....	131
Managing System Parameters	140
Managing VPS Resources Configuration	146

What are Resource Control Parameters?

The system administrator controls the resources available to a Virtual Private Server through a set of resource management parameters. All these parameters are defined either in the Virtuozzo global configuration file (`/etc/sysconfig/vz`), or in the respective VPS configuration files (`/etc/sysconfig/vz-scripts/$VPSID`), or in both. You can set them by manually editing the corresponding configuration files, by using the Virtuozzo command-line utilities, or thru Virtuozzo Management Console. These parameters can be divided into the disk, network, CPU, and system categories. The table below summarizes these groups:

Group	Description	Parameter names	Explained in
Disk	This group of parameters determines disk quota in Virtuozzo. The Virtuozzo disk quota is realized on two levels: the per-VPS level and the per-user/group level. You can turn on/off disk quota on any level and configure its settings.	<code>DISK_QUOTA</code> , <code>DISKSPACE</code> , <code>DISKINODES</code> , <code>QUOTATIME</code> , <code>QUOTAUGIDLIMIT</code>	Managing Disk Quotas
Network	This group of parameters determines the management of network bandwidth available to different VPSs (network shaping). You can turn on/off network shaping and configure the settings for different VPSs.	<code>TRAFFIC_SHAPING</code> , <code>BANDWIDTH</code> , <code>TOTALRATE</code> , <code>RATE</code> , <code>RATEBOUND</code>	Managing Network Accounting and Bandwidth

CPU	This group of parameters defines the CPU time different VPSs are guaranteed to receive.	VE0CPUUNITS, CPUUNITS	Managing CPU Share
System	This group of parameters defines various aspects of using system memory, TCP sockets, IP packets and like parameters by different VPSs.	avnumproc, numproc, numtcpsock, numothersock, vmguarpages, kmemsize, tcpsndbuf, tcprcvbuf, othersockbuf, dgramrcvbuf, oomguarpages, lockedpages, shmpages, privvmpages, physpages, numfile, numflock, numpty, numsiginfo, dcachesize, numiptent	Managing System Parameters

Managing Disk Quotas

This section explains what disk quotas are, defines disk quota parameters, and describes how to perform disk quota related operations:

- Turning on and off per-VPS (first-level) disk quotas;
- Setting up first-level disk quota parameters for a Virtual Private Server;
- Turning on and off per-user and per-group (second-level) disk quotas inside a Virtual Private Server;
- Setting up second-level quotas for a user or for a group;
- Checking disk quota statistics;
- Cleaning up Virtual Private Servers in certain cases.

What are Disk Quotas?

Disk quotas enable system administrators to control the size of Linux file systems by limiting the amount of disk space and the number of inodes a Virtual Private Server can use. These quotas are known as per-VPS quotas or first-level quotas in Virtuozzo. In addition, Virtuozzo enables the Virtual Private Server administrator to limit disk space and the number of inodes that individual users and groups in that VPS can use. These quotas are called per-user and per-group quotas or second-level quotas in Virtuozzo.

By default, Virtuozzo has first-level quotas enabled (which is defined in the Virtuozzo global configuration file), whereas second-level quotas must be turned on for each Virtual Private Server separately (in the corresponding VPS configuration files). It is impossible to turn on second-level disk quotas for a Virtual Private Server if first-level disk quotas are off for that Virtual Private Server.

The disk quota block size in Virtuozzo is always 1024 bytes. It may differ from the block size of the underlying file system.

Virtuozzo keeps quota usage statistics and limits in `/var/vzquota/quota.<vpsid>` - a special quota file. The quota file has a special flag indicating whether the file is “dirty”. The file becomes dirty when its contents become inconsistent with the real VPS usage. This means that when the disk space or inodes usage changes during the VPS operation, these statistics are not automatically synchronized with the quota file, the file just gets the “dirty” flag. They are synchronized only when the VPS is stopped or when the HN is shut down. After synchronization, the “dirty” flag is removed. If the Hardware Node has been incorrectly brought down (for example, the power switch was hit), the file remains “dirty”, and the quota is re-initialized on the next VPS startup. This operation may noticeably increase the Node startup time. Thus, it is highly recommended to shut down the Hardware Node properly.

Disk Quota Parameters

The table below summarizes the disk quota parameters that you can control. The File column indicates whether the parameter is defined in the Virtuozzo global configuration file (G), in the VPS configuration files (V), or it is defined in the global configuration file but can be overridden in a separate VPS configuration file (GV).

Parameter	Description	File
<code>disk_quota</code>	Indicates whether first-level quotas are on or off for all VPSs or for a separate VPS.	GV
<code>diskspace</code>	Total size of disk space the VPS may consume, in 1-Kb blocks.	V
<code>diskinodes</code>	Total number of disk inodes (files, directories, and symbolic links) the Virtual Private Server can allocate.	V
<code>quotatime</code>	The grace period for the disk quota overusage defined in seconds. The Virtual Private Server is allowed to temporarily exceed its quota soft limits for no more than the QUOTATIME period.	V
<code>quotauidlimit</code>	Number of user/group IDs allowed for the VPS internal disk quota. If set to 0, the UID/GID quota will not be enabled.	V

Turning On and Off Per-VPS Disk Quotas

The parameter that defines whether to use first-level disk quotas is `DISK_QUOTA` in the Virtuozzo global configuration file (`/etc/sysconfig/vz`). By setting it to “no”, you will disable Virtuozzo quotas completely.

This parameter can be specified in the Virtual Private Server configuration file (`/etc/sysconfig/vz-scripts/<veid>.conf`) as well. In this case its value will take precedence of the one specified in the global configuration file. If you intend to have a mixture of Virtual Private Servers with quotas turned on and off, it is recommended to set the `DISK_QUOTA` value to “yes” in the global configuration file and to “no” in the configuration file of that VPS which does not need quotas.

The session below illustrates a scenario when first-level quotas are on by default and are turned off for Virtual Private Server 101:

```
[checking that quota is on]
# grep DISK_QUOTA /etc/sysconfig/vz
DISK_QUOTA=yes

[checking available space on /vz partition]
# df /vz
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/sda2              8957295    1421982   7023242  17% /vz

[editing VPS configuration file to add DISK_QUOTA=no]
# vi /etc/sysconfig/vz-scripts/101.conf

[checking that quota is off for VPS 101]
# grep DISK_QUOTA /etc/sysconfig/vz-scripts/101.conf
DISK_QUOTA=no

# vzctl start 101
Starting VPS ...
VPS is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: ve101
VPS start in progress...
# vzctl exec 101 df
Filesystem            1k-blocks      Used Available Use% Mounted on
vzfs                  8282373     747060   7023242  10% /
```

As the above example shows, the only disk space limit a Virtual Private Server with the quotas turned off has is the available space and inodes on the partition where the VPS private area resides.

To view and/or change the `DISK_QUOTA` parameter using VZMC, do the following:

- 1 Select the Configuration item under the needed Node in the VZMC left pane:

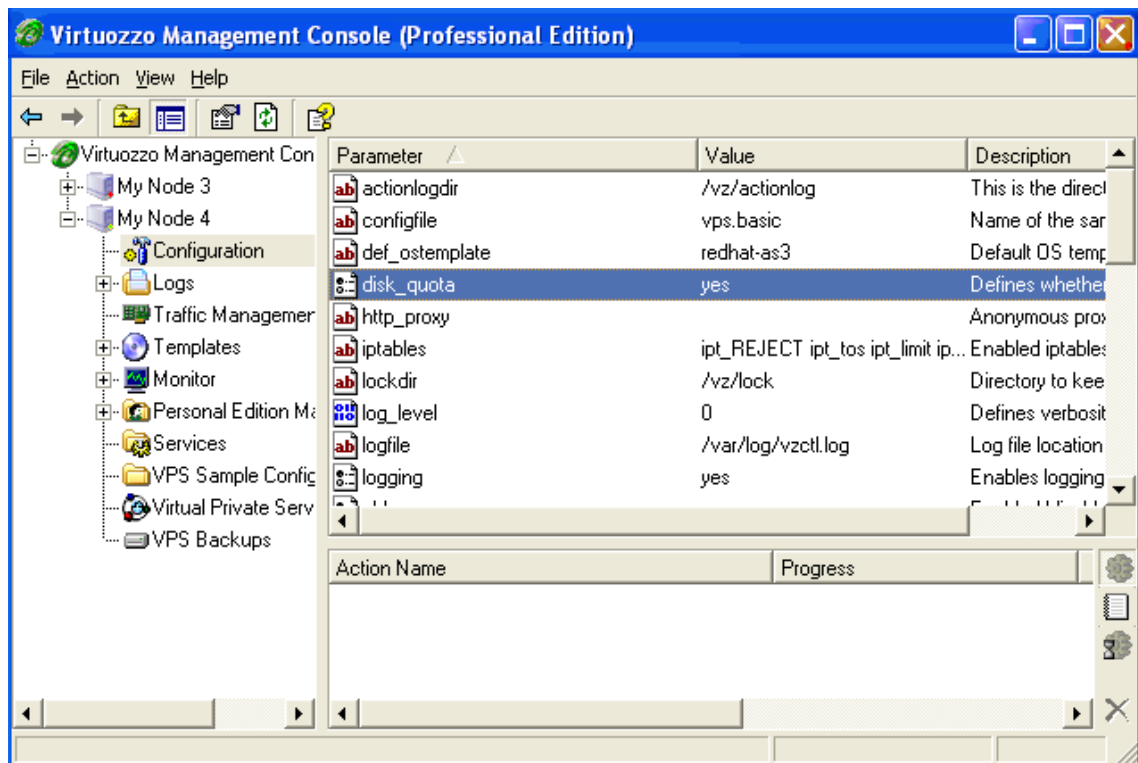


Figure 23: VZMC - Turning First-Level Disk Quota On and Off

- 2 Double-click on the `disk_quota` parameter in the right pane.
- 3 Select the needed value and press OK.

VZMC does not let you enable/disable disk quotas for separate Virtual Private Servers, thus overriding the global setting. If the first-level quotas are on by default, there is no way to rescind the calculation of quota data for a VPS by means of VZMC. However, you can allow this VPS to have an almost unlimited disk space and the number of inodes by doing the following:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.

- 2 Click the QoS tab and expand Disk Quota parameters:

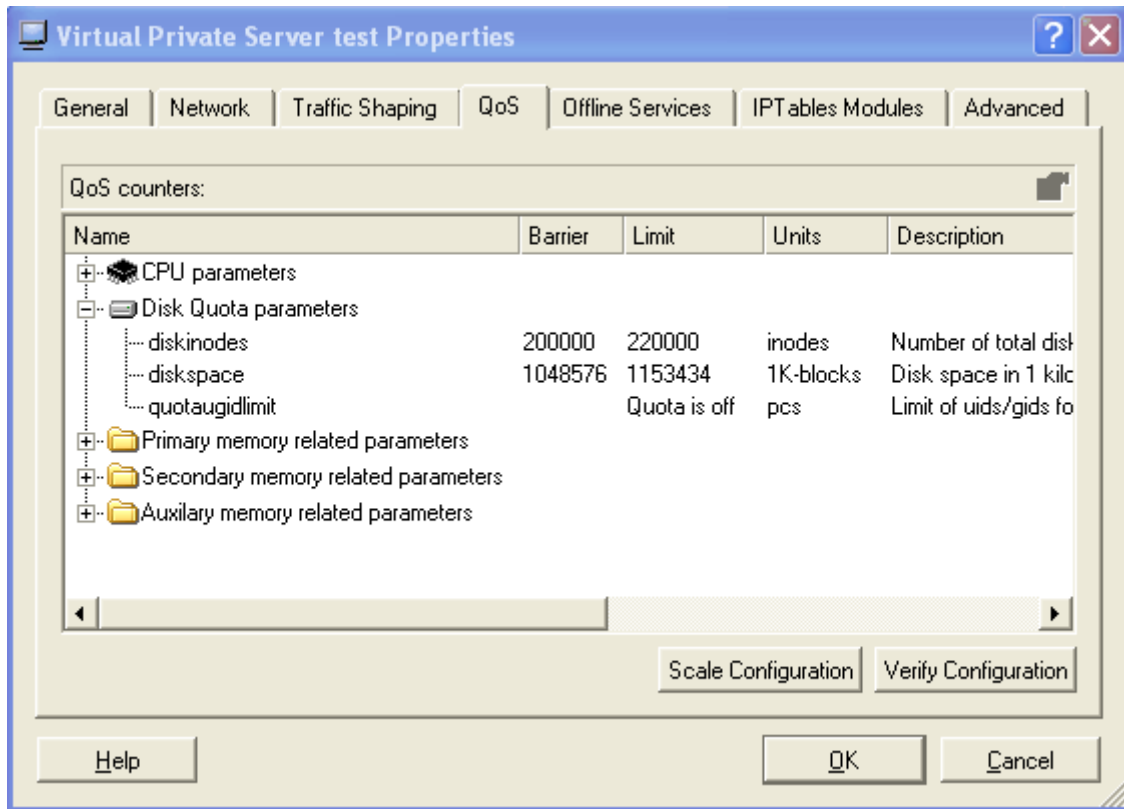


Figure 24: VZMC - VPS Disk Quota Parameters

- 3 Double-click the `diskinodes` parameter, and toggle off/on the **Not limited** checkbox depending on whether you want to limit or not the number of disk inodes for the given VPS, respectively.
- 4 Press OK.
- 5 If necessary, repeat Steps 3 and 4 for the `diskspace` parameter to limit the disk space for the given VPS or allow it to have unlimited disk space.

Note: You must change the `DISK_QUOTA` parameter in the global VirtuoZZo configuration file only when all Virtual Private Servers are stopped, and in the VPS configuration file – only when the corresponding VPS is stopped. Otherwise, the configuration may prove inconsistent with the real quota usage, and this can interfere with the normal Hardware Node operation.

Setting Up Per-VPS Disk Quota Parameters

Three parameters determine how much disk space and inodes a Virtual Private Server can use. These parameters are specified in the Virtual Private Server configuration file:

DISKSPACE Total size of disk space that can be consumed by the Virtual Private Server in 1-Kb blocks. When the space used by the Virtual Private Server hits the soft limit, the VPS can allocate additional disk space up to the hard limit during the grace period specified by the `QUOTATIME` parameter.

DISKINODES	Total number of disk inodes (files, directories, and symbolic links) the Virtual Private Server can allocate. When the number of inodes used by the Virtual Private Server hits the soft limit, the VPS can create additional file entries up to the hard limit during the grace period specified by the QUOTATIME parameter.
QUOTATIME	The grace period of the disk quota specified in seconds. The Virtual Private Server is allowed to temporarily exceed the soft limit values for the disk space and disk inodes quotas for no more than the period specified by this parameter.

The first two parameters have both soft and hard limits (or, simply, barriers and limits). The hard limit is the limit that cannot be exceeded under any circumstances. The soft limit can be exceeded up to the hard limit, but as soon as the grace period expires, the additional disk space or inodes allocations will fail. Barriers and limits are separated by colons (":") in Virtual Private Server configuration files and in the command line.

The following session sets the disk space available to Virtual Private Server 101 to approximately 1Gb and allows the VPS to allocate up to 90,000 inodes. The grace period for the quotas is set to ten minutes:

```
# vzctl set 101 --diskspace 1000000:1100000 --save
Saved parameters for VPS 101
# vzctl set 101 --diskinodes 90000:91000 --save
Saved parameters for VPS 101
# vzctl set 101 --quotatime 600 --save
Saved parameters for VPS 101
# vzctl exec 101 df
Filesystem            1k-blocks      Used Available Use% Mounted on
vzfs                   1000000      747066    252934   75% /
# vzctl exec 101 stat -f /
File: "/"
  ID: 0          0      Namelen: 255      Type: UNKNOWN (0x565a4653)
Blocks: Total: 1000000  Free: 252934  Available: 252934  Size: 1024
Inodes: Total: 90000   Free: 9594
```

It is possible to change the first-level disk quota parameters for a running Virtual Private Server. The changes will take effect immediately. If you do not want your changes to persist till the next Virtual Private Server startup, do not use the `--save` switch.

To set up per-VPS disk quota parameters using VZMC, do the following:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.
- 2 Click the **QoS** tab and expand **Disk Quota** parameters.

- 3 Double-click the `diskinodes` parameter, and enter the soft limit and hard limit values for this parameter in the fields provided. For example:

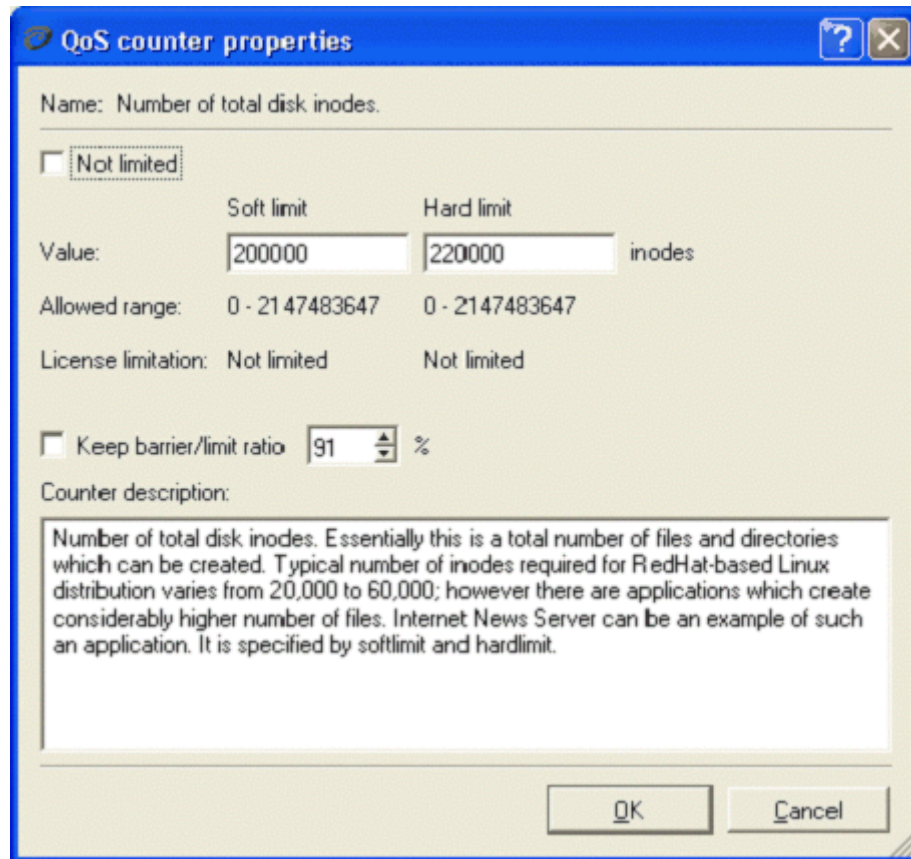


Figure 25: VZMC - Setting Up VPS Disk Quota

The hard limit is the limit that cannot be exceeded under any circumstances. The soft limit can be exceeded up to the hard limit, but as soon as the grace period expires, the additional disk space or inodes allocations will fail.

- 4 Press OK.
- 5 If necessary, repeat Steps 3 and 4 for the `diskspace` parameter to define the disk space quota for the given VPS.

Note: Currently there is no way to define the `quotatime` parameter by means of the VZMC interface.

Turning On and Off Second-Level Quotas for Virtual Private Server

The parameter that controls the second-level disk quotas is `QUOTAUGIDLIMIT` in the VPS configuration file. By default, the value of this parameter is zero and this corresponds to disabled per-user/group quotas.

If you assign a non-zero value to the `QUOTAUGIDLIMIT` parameter, this action brings about the two following results:

- 1 Second-level (per-user and per-group) disk quotas are enabled for the given Virtual Private Server;
- 2 The value that you assign to this parameter will be the limit for the number of file owners and groups of this VPS, including Linux system users. Note that you will theoretically be able to create extra users of this VPS, but if the number of file owners inside the VPS has already reached the limit, these users will not be able to own files.

Enabling per-user/group quotas for a Virtual Private Server requires restarting the VPS. The value for it should be carefully chosen; the bigger value you set, the bigger kernel memory overhead this Virtual Private Server creates. This value must be greater than or equal to the number of entries in the VPS `/etc/passwd` and `/etc/group` files. Taking into account that a newly created Red Hat Linux-based VPS has about 80 entries in total, the typical value would be 100. However, for Virtual Private Servers with a large number of users this value may be increased.

The session below turns on second-level quotas for Virtual Private Server 101:

```
# vzctl set 101 --quotaugidlimit 100 --save
Unable to apply new quota values: ugid quota not initialized
Saved parameters for VPS 101
# vzctl stop 101; vzctl start 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
Starting VPS ...
VPS is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: ve101
VPS start in progress...
```

In VZMC, Virtuozzo second-level disk quotas are controlled in the window that you may access by performing the following actions:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.
- 2 Click the **QoS** tab and expand **Disk Quota** parameters.

- 3 Double-click the `quotauidlimit` parameter:

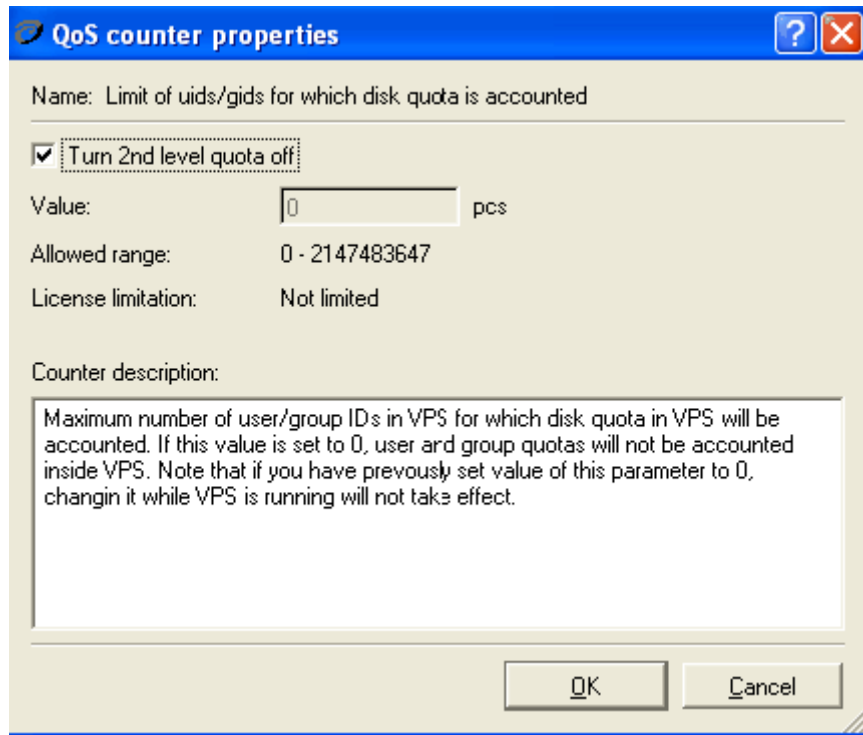


Figure 26: VZMC - Turning Second-Level Disk Quota On and Off

- 4 Enter the right settings for you and press OK. In the example above, the second-level quotas are off for the corresponding VPS.

Setting Up Second-Level Disk Quota Parameters

Virtuozzo provides the standard Linux quota package for working inside Virtual Private Servers:

```
# vzctl exec 101 rpm -q quota
quota-3.03-1.1.swsoft
```

This command shows that the quota package installed in the Virtual Private Server is built and shipped by SWsoft. Use the utilities from this package (as is prescribed in your Linux manual) to set Virtuozzo second-level quotas for the given VPS. For example:

```
# ssh vps101
root@vel01's password:
Last login: Sat Jul 5 00:37:07 2003 from 10.100.40.18
[root@vps101 root]# edquota root
Disk quotas for user root (uid 0):
  Filesystem  blocks      soft   hard   inodes      soft   hard
  /dev/vzfs   38216      50000  60000   45454      70000  70000
[root@vel01 root]# repquota -a
*** Report for user quotas on device /dev/vzfs
Block grace time: 00:00; Inode grace time: 00:00
      Block limits                File limits
User      used    soft   hard  grace    used    soft   hard  grace
-----
root      --    38218  50000  60000          45453  70000  70000
[the rest of repquota output is skipped]

[root@vel01 root]# dd if=/dev/zero of=test
dd: writing to `test': Disk quota exceeded
23473+0 records in
23472+0 records out
[root@vel01 root]# repquota -a
*** Report for user quotas on device /dev/vzfs
Block grace time: 00:00; Inode grace time: 00:00
      Block limits                File limits
User      used    soft   hard  grace    used    soft   hard  grace
-----
root      +-    50001  50000  60000   none    45454  70000  70000
[the rest of repquota output is skipped]
```

The above example shows the session when the root user has the disk space quota set to the hard limit of 60,000 1Kb blocks and to the soft limit of 50,000 1Kb blocks; both hard and soft limits for the number of inodes are set to 70,000.

It is also possible to set the grace period separately for block limits and inodes limits with the help of the `/usr/sbin/setquota` command. For more information on using the utilities from the quota package, please consult the system administration guide shipped with your Linux distribution or online manual pages included in the package.

Virtuozzo Management Console also provides means for setting up second-level disk quotas in Virtuozzo. You should perform the following steps:

- 1 Open the needed Virtual Private Server manager window by double-clicking on the corresponding Virtual Private Server line in the right pane of the VZMC window;
- 2 Expand the Users and Groups item in the left pane of the VPS manager window;

- 3 Select either the Groups folder or the Users folder to see the list of VPS registered groups/users in the right pane:

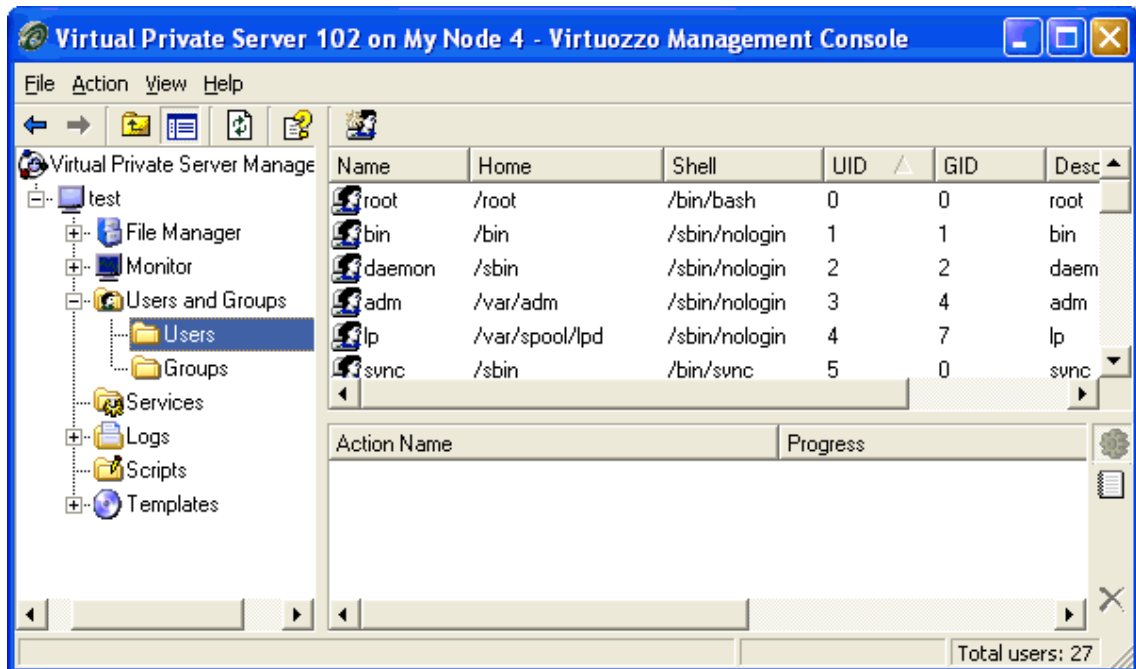


Figure 27: VZMC - Viewing List of VPS Users

- 4 Double-click the name of the group/user for whom you want to set up the quota parameters. The group/user Properties window appears.

- 5 Click the Disk Quota tab in this window:

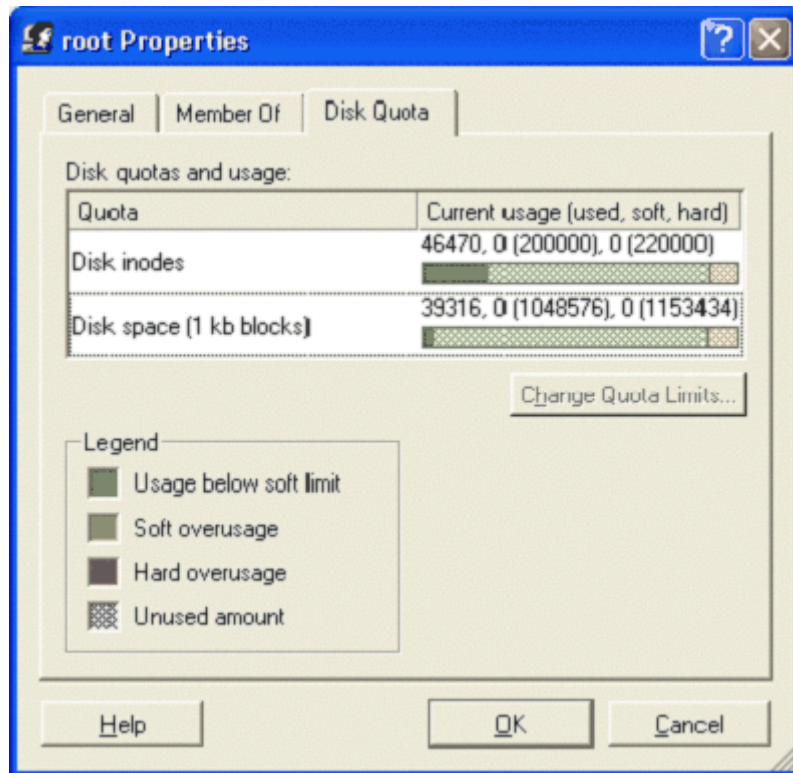


Figure 28: VZMC - Setting Up Per-User Disk Quota

- 6 Select the needed quota parameter and click Change Quota Limits;
- 7 In the window that appears, enter the quota settings of your choice for the current group/user.
- 8 Press OK to close the Second Level Disk Quota window; then press OK to close the group/user Properties window.

Checking Quota Status

As the Hardware Node administrator, you can check the quota status for any Virtual Private Server with the `vzquota stat` and `vzquota show` commands. The first command reports the status from the kernel and shall be used for running Virtual Private Servers. The second command reports the status from the quota file (located at `/var/vzquota/quota.<vpsid>`) and shall be used for stopped Virtual Private Servers. Both commands have the same output format.

The session below shows a partial output of VPS 101 quota statistics:

```
# vzquota stat 101 -t
  resource      usage      softlimit      hardlimit      grace
  1k-blocks     38281      1000000       1100000
  inodes        45703      90000         91000
User/group quota: on,active
Ugids: loaded 34, total 34, limit 100
Ugid limit was exceeded: no

User/group grace times and quotafile flags:
  type block_exp_time inode_exp_time  dqf_flags
  user                               0h
  group                             0h

User/group objects:
ID   type   resource  usage  softlimit  hardlimit  grace status
0    user  1k-blocks 38220   50000     60000     loaded
0    user  inodes   45453   70000     70000     loaded
[the rest is skipped]
```

The first three lines of the output show the status of first-level disk quotas for the Virtual Private Server. The rest of the output displays statistics for user/group quotas and has separate lines for each user and group ID existing in the system.

If you do not need the second-level quota statistics, you can omit the `-t` switch from the `vzquota` command line.

To check the first-level quota status for a VPS in VZMC, you should:

- 1 Open the needed Virtual Private Server manager window by double-clicking on the corresponding Virtual Private Server line in the right pane of the VZMC window;
- 2 Expand the **Monitor** item and select the **Quotas and Usage** folder.

You can see the first-level quota statistics for the current VPS in the right pane of the window:

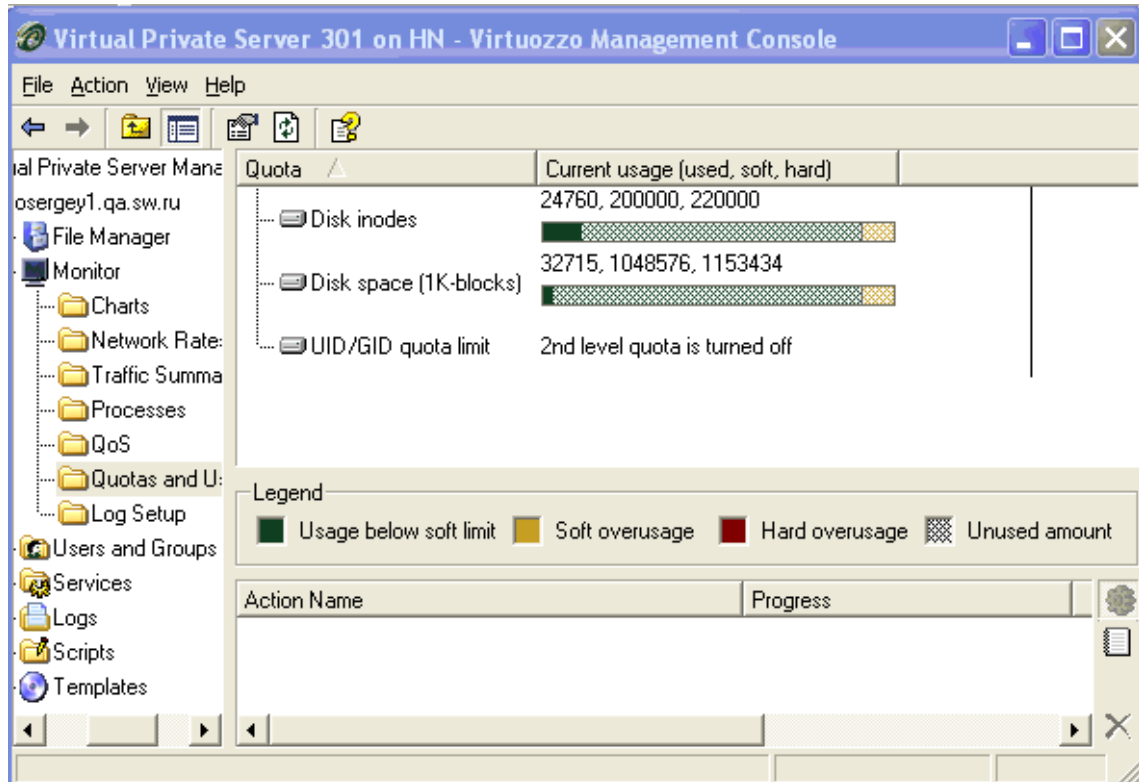


Figure 29: VZMC - Viewing VPS Disk Quota Statistics

To check the second-level disk quota parameters for any group or user of the given VPS, perform Steps 1 thru 5 as is indicated in the previous section.

Cleaning Up Virtual Private Servers

The first-level quota assigned to this or that Virtual Private Server essentially shows how much space may be occupied by the VPS *private* files, i.e. not by the OS or common applications files. The real OS and application files reside in the `/vz/template` directory on the Hardware Node and practically do not add up to the VPS quota (except for the symlinks to them located inside the VPS and occupying insignificant space).

However, there are situations when one and the same application or application update is installed not as a template, but separately inside each and every VPS. A good example of this is the CPanel application with its robust auto-update features. If a certain version of CPanel is installed in a number of VPSs, and then an update is released, CPanel automatically updates itself in all these VPSs, thus creating a vast amount of identical files (not symlinks already) throughout the VPSs. These files tell dramatically on the VPS quotas, which may be avoided by putting all the identical files to the Hardware Node template area and creating symlinks instead of real files inside the affected VPSs.

The problem like the one described above can be solved in two ways:

- 1 A special subarea is created inside the Hardware Node template area for housing the files identical among multiple VPSs with the help of the `vzcache` utility.

- 2** If the application or application update installed directly into one or more VPSs has a corresponding application template or template update installed on the Hardware Node, the real files inside the VPS(s) are replaced with symlinks to the template files on the Node with the help of the `vzpkglink` utility.

Moving VPS Files to Cache Area on Hardware Node

We will illustrate the effect produced by `vzcache` by copying one and the same huge dummy file into two VPSs. First, let us learn the disk space occupied by the whole `/vz` partition and by the two VPSs:

```
# df /vz
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda3              13756796    1348292   11622123   11% /vz
# vzctl exec 101 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                   1048576      22311    1026265    3% /
# vzctl exec 102 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                   1048576      22311    1026265    3% /
```

After that, we copy the dummy file, which is around 600 Mb in size, to the root of these VPSs:

```
# cp foo /vz/root/101
# cp foo /vz/root/102
```

Now, check the disk space once again:

```
# df /vz
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda3              13756796    2569060   10401355   20% /vz
# vzctl exec 101 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                   1048576     632430     416146   61% /
# vzctl exec 102 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                   1048576     632430     416146   61% /
```

We see that around 600 Mb has been added to the space occupied by each VPS and, consequently, around 1.2 Gb has been added to the space used on the `/vz` partition. Now, it's time to resort to `vzcache` to get rid of identical files inside the VPSs:

```
# vzcache 101 102
Optimizing private area for VE 101 ...
[Optimization messages skipped...]
Optimizing private area for VE 102 ...
[Optimization messages skipped...]
Done.
VE 101                      78 regular files
VE 102                      78 regular files

Caching summary
VEID (cached/total)        Files      Size
101                        70/156     616181K/616189K
102                        70/156     616181K/616189K

Cache usage
                           Files      Size
dhcp0-84.sw.ru-2004030316237  70       616181K
Creating cache in /vz/template/vzcaches/dhcp0-84.sw.ru-2004030316237
```

In the **Caching summary** table, `vzcache` outputs the number and total size of regular files (not symlinks) it has found in the specified VPSs and of the identical files to be cached. In our case, it happened to be 70 identical files in VPSs 101 and 102 amounting to the size of 616,181 Kb. All these files have been moved to the Hardware Node directory specified in the last line of the output, and symlinks to them have been created inside the VPSs. So, the final look at the disk space usage:

```
# df /vz
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda3             13756796    1953053   11017362   16% /vz
# vzctl exec 101 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                  1048576      15105    1033471    2% /
# vzctl exec 102 df
Filesystem            1K-blocks      Used Available Use% Mounted on
vzfs                  1048576      15138    1033438    2% /
```

shows that both the Hardware Node and the Virtual Private Servers have each gained more than 600 Mb of disk space. In real life, the disk space is gained by caching not one huge file in two VPSs but a number of identical files across many VPSs.

The operation of the `vzcache` utility may be customized to a certain extent by using `vzcache` command line switches (see the [Virtuozzo Reference Guide](#) for details).

Associating VPS Files With Application Templates

It may often happen that a security update should immediately be applied to a package installed as a template on the Node and added to a number of VPSs hosted there. However, it takes certain time to prepare a template update, so the Hardware Node and/or VPS administrators are not inclined to wait for it and they install the original security update directly inside the VPSs. As to the template update, it becomes available a few days afterwards. In other cases, a VPS administrator might not know that there is a certain template installed on the Hardware Node, so they install the corresponding application directly inside their VPS.

To eliminate cluttering up the VPS disk space with application files that are present as part of an application template on the Hardware Node, the `vzpkglink` utility is used. First, check if the VPS files are compatible with the template version installed on the Node. For example:

```
# vzpkglink -t -vv 101 openssl/20041118
```

If this test performs successfully, you can drop the `-t` switch and replace the `openssl` files inside VPS 101 with symlinks to these files in the `/vz/template/openssl` directory on the Hardware Node:

```
vzpkglink -vv 101 openssl/20041118
```

Issuing the `vzpkgls 101` command now will let you ensure that the `openssl` template has been added to the VPS configuration file.

Backing Up and Restoring Caches

If you are backing up and restoring Virtual Private Servers with the help of the `vzbackup` utility, it does not back up and restore VPS caches by default. However, VPS caches do contain VPS private files, therefore you may want to back them up as well. This is done with the help of a special switch of the `vzbackup` utility, for example:

```
# vzbackup --vzcache 192.168.20.20 -e 101
```

This command will back up all the cache areas VPS 101 is appended to, but not VPS 101 itself. In order to back up both the VPSs and their caches, you will have to back up the whole Node:

```
# vzbackup 192.168.20.20
```

The restoring of VPS caches is performed in much the same way: the `--vzcache` option should be used with the `vzrestore` utility. However, the restoring of the whole Node will not restore the VPS caches by default, and you should explicitly provide the `--vacache` option in all cases.

Detaching Virtual Private Server From Hardware Node Cache Directory

Whereas the `vzcache` utility helps effectively gain disk space both in the Hardware Node and within Virtual Private Servers, there may be situations when it is necessary to detach a Virtual Private Server from its cache and copy the cached files back to the VPS private area. A typical example of this is migrating a VPS to another Hardware Node. The migration is not possible if there are links in the VPS private area pointing to the `/vz/template/vzcaches` directory on the Hardware Node.

To copy the cached files back to the VPS private area, the `vzuncache` utility is used:

```
[root@dhcp0-84 root]# vzuncache 101 -a
[Optimization messages skipped...]
VE 101                               53 magic symlinks to convert

VE 101 will be detached from the following caches:
Cache name                               Size
dhcp0-84.sw.ru-2004030316237             607972K
```

Now, VPS 101 can safely be migrated to another Hardware Node. Note that unlike `vzcache`, the `vzuncache` utility shall be called for only one VPS at a time. The `-a` switch tells the utility to detach the VPS from all the cache directories specified in its configuration file as the value of the `VZCACHE` parameter.

Managing CPU Share

The current section explains the CPU resource parameters (CPU share) that you can configure and monitor for each Virtual Private Server.

The table below provides the name and the description for the CPU parameters. The **File** column indicates whether the parameter is defined in the Virtuozone global configuration file (G) or in the VPS configuration files (V).

Parameter	Description	File
ve0cpuunits	This is a positive integer number that determines the minimal guaranteed share of the CPU time Virtual Private Server 0 (the Hardware Node itself) will receive. It is recommended to set the value of this parameter to be 5-10% of the power of the Hardware Node.	G
cpuunits	This is a positive integer number that determines the minimal guaranteed share of the CPU time the corresponding Virtual Private Server will receive.	V
cpulimit	This is a positive number indicating the CPU time in per cent the V corresponding VPS is not allowed to exceed.	V

The Virtuoazzo 3.0 CPU resource control utilities allow you to guarantee any Virtual Private Server the amount of CPU time this Virtual Private Server receives. The Virtual Private Server can consume more than the guaranteed value if there are no other Virtual Private Servers competing for the CPU and the `cpulimit` parameter is not defined.

Note: The CPU time shares and limits are calculated on the basis of a one-second period. Thus, for example, if a VPS is not allowed to receive more than 50% of the CPU time, it will be able to receive no more than half a second each second.

To get a view of the optimal share to be assigned to a Virtual Private Server, check the current Hardware Node CPU utilization:

```
# vzcpucheck
Current CPU utilization: 5166
Power of the node: 73072.5
```

The output of this command displays the total number of the so-called CPU units consumed by all running Virtual Private Servers and Hardware Node processes. This number is calculated by Virtuoazzo with the help of a special algorithm. The above example illustrates the situation when the Hardware Node is underused. In other words, the running Virtual Private Servers receive more CPU time than was guaranteed to them.

In the following example, Virtual Private Server 102 is guaranteed to receive about 2% of the CPU time even if the Hardware Node is fully used, or in other words, if the current CPU utilization equals the power of the Node. Besides, VPS 102 will not receive more than 4% of the CPU time even if the CPU is not fully loaded:

```
# vzctl set 102 --cpuunits 1500 --cpulimit 4 --save
Saved parameters for VE 102
# vzctl start 102
Starting VE ...
VE is mounted
Adding IP address(es): 192.168.1.102
VE start in progress...
# vzcpucheck
Current CPU utilization: 6667
Power of the node: 73072.5
```

Virtual Private Server 102 will receive from 2 to 4% of the Hardware Node CPU time unless the Hardware Node is overcommitted, i.e. the running Virtual Private Servers have been promised more CPU units than the power of the Hardware Node. In this case the VPS might get less than 2 per cent.

To view and/or change the `VE0CPUUNITS` parameter using VZMC, do the following:

- 1 Select the **Configuration** item under the needed Node in the VZMC left pane.
- 2 Double-click on the `ve0cpuunits` parameter in the right pane.
- 3 Enter the needed value and press **OK**.

To view and/or change the `CPUUNITS` or `CPULIMIT` parameter for separate Virtual Private Servers, do the following:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.
- 2 Click the **QoS** tab and expand **CPU parameters**.

- 3 Double-click the corresponding parameter, and, if necessary, enter the right value for the given VPS.
- 4 Press OK.

Managing Network Accounting and Bandwidth

This section explains how to perform the following tasks:

- Setting up network classes;
- Viewing network traffic statistics;
- Turning on and off network bandwidth management;
- Setting up the bandwidth limit for a Virtual Private Server.

Network Traffic Parameters

The table below summarizes the network traffic parameters that you can control. The File column indicates whether the parameter is defined in the Virtuozzo global configuration file (G), in the VPS configuration files (V), or it is defined in the global configuration file but can be overridden in a separate VPS configuration file (GV).

Parameter	Description	File
<code>traffic_shaping</code>	If set to “yes”, traffic limitations for outgoing traffic are set for Virtual Private Servers. The default is “no”.	G
<code>bandwidth</code>	This parameter lists all the network adapters installed on the Hardware Node and their bandwidth.	G
<code>totalrate</code>	This parameter defines the bandwidth to be allocated for each and every network class. It is active if traffic shaping is turned on.	G
<code>rate</code>	If traffic shaping is turned on, this parameter specifies the bandwidth guarantee for any VPS.	GV
<code>ratebound</code>	If this parameter is set to “yes”, the bandwidth guarantee (the global rate parameter) is also the limit for the VPS, and the VPS cannot borrow the bandwidth from the TOTALRATE bandwidth pool.	V

Note: In old configuration files, there may remain the `traffic_accounting` parameter in the global configuration file. It is outdated in the current Virtuozzo version, as traffic accounting is always enabled now.

Configuring Network Classes

Virtuozzo allows you to track the inbound and outbound network traffic as well as to shape the outgoing traffic for Virtual Private Servers. In order to provide the ability to distinguish between domestic and international traffic, a concept of network classes is introduced. It is important to fully understand this notion, because network classes IDs are used in the values of some network traffic parameters. A network class is a range of IP addresses for which Virtuozzo counts and shapes the traffic.

Virtuozzo can have up to 15 different network classes specified. Each class can contain one or more IP address ranges. It is possible to have different bandwidth shaping settings for each class.

Classes are specified in the `/etc/sysconfig/vz-scripts/networks_classes` file. The file is in the ASCII format, and all empty lines and lines starting with the '#' sign are ignored. Other lines have the following format:

```
<class_id> <ip_address>/<prefix_length>
```

where `<class_id>` defines the network class id, and the `<ip_address>/<prefix_length>` pair defines the range of IP addresses for this class. There may be several lines for each class.

Classes 0 and 1 have special meanings. Class 0 defines the IP address range for which no accounting is done. Usually, it corresponds to the Hardware Node subnet (the Node itself and its VPSs). Setting up Class 0 is not required; however, its correct setup improves performance.

Class 1 is defined by Virtuozzo to match any IP address. It must be always present in the network classes definition file. Therefore, it is suggested not to change the default line

```
1 0.0.0.0/0
```

Other Classes should be defined after Class 1. They represent exceptions from the "matching-everything" rule of Class 1. The example below illustrates a possible configuration of the network classes definition file:

```
# HW node VPSs networks
0 192.168.0.0/16

# any IP (all traffic)
1 0.0.0.0/0

# class 2 - addresses for the "foreign" traffic
2 10.0.0.0/8
2 11.0.0.0/8

# inside "foreign" network there
# is a hole belonging to "local" traffic
1 10.10.16.0/24
```

In this example, the IP addresses in the range of 192.168.0.0 to 192.168.255.255 are treated as Class 0 addresses and no accounting is done for the traffic from Virtual Private Servers destined to these addresses.

Class 2 matches addresses in two ranges: from 10.0.0.0 to 10.255.255.255 and from 11.0.0.0 to 11.255.255.255 with the exception of addresses in the sub-range of 10.10.16.0 to 10.10.16.255, which are treated as Class 1. All other IP addresses belong to Class 1. As far as the Class 2 addresses in this example are used for foreign routing, the Class 1 addresses are used for local (domestic) routing, by the exclusion method.

To set up network classes by means of Virtuozzo Management Console, you should:

- 1 Select the **Traffic Management** item under the necessary Node in the left pane of the VZMC window;
- 2 Follow the [Click here to change traffic accounting settings](#) link in the right pane;
- 3 On the **Accounting** tab of the **Traffic Accounting and Shaping** window, press the **New IP range** button to display the following window:

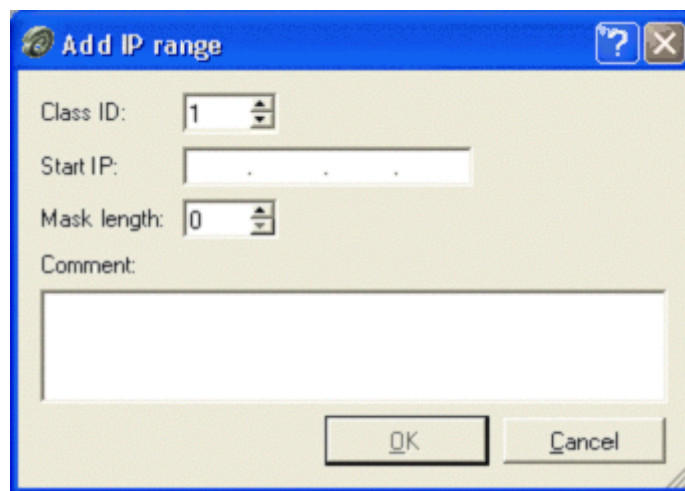


Figure 30: VZMC - Configuring Network Classes

- 4 Fill in the fields provided (the first three fields are mandatory) and press **OK**.

Now you should have another network class defined in your network classes configuration file. To edit or delete the existing classes, use the corresponding buttons on the **Accounting** tab in the **Traffic Accounting and Shaping** window.

Viewing Network Traffic Statistics

Virtuozzo allows you to view the current network traffic statistics with the help of the `vznetstat` command. The session below shows the traffic statistics for Virtual Private Server 101:

```
# vznetstat -v 101
VEID  Net.Class  Input(bytes) Input(pkts)  Output(bytes) Output(pkts)
101   1          2202448     19527       9081832     19584
101   2          0           0           0           0
```

In this case, around 2Mb of data were uploaded to the Virtual Private Server and about 9Mb were downloaded from it. All the traffic matches the definition of Class 1 and no data was exchanged with any hosts from Class 2 networks.

Without specifying Virtual Private Server ID with the `-v` parameter, the command will display the statistics for all running Virtual Private Servers.

In VZMC, you can also view the current network traffic statistics. Just select the **Traffic Management** item under the necessary Node in the left pane. For example:

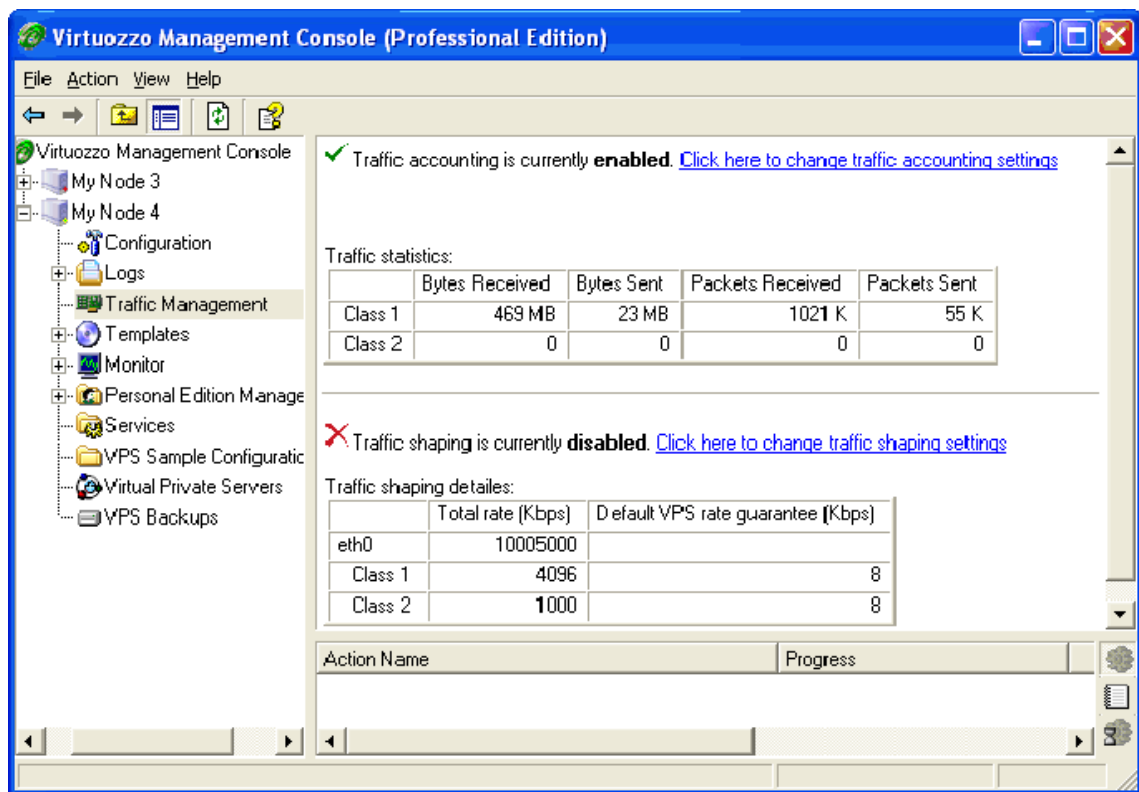


Figure 31: VZMC - Viewing Network Traffic Statistics

If you wish to see the network traffic statistics for a separate VPS, do the following:

- 1 Open the needed Private Server manager window by double-clicking on the corresponding Virtual Private Server line in the right pane of the VZMC window;
- 2 Expand the **Monitor** item and select the **Network** folder. You can now see the network traffic statistics for the given VPS in the right pane of the window.

Turning On and Off Network Bandwidth Management

Traffic shaping also known as network bandwidth management allows you to control what network bandwidth a Virtual Private Server receives for outgoing traffic. Traffic shaping is off by default in Virtuozzo and is controlled by the `TRAFFIC_SHAPING` variable in the Virtuozzo global configuration file (`/etc/sysconfig/vz`).

Note: VPS incoming traffic cannot be controlled in Virtuozzo 3.0.

In order to turn traffic shaping on, you have to complete the following steps:

- Set the value of `TRAFFIC_SHAPING` to “yes” in the Virtuozzo global configuration file;
- Correctly set up the `BANDWIDTH` and `TOTALRATE` parameters values;
- Start traffic shaping with the `/etc/init.d/vz shaperon` command.

The `BANDWIDTH` variable is used for specifying the network rate (in kilobits per second) of available network adapters. By default, it is set to “`eth0:102400`”, which corresponds to a 100Mb/s Fast Ethernet card. If your Hardware Node has more network adapters installed, you need to update this variable to list all the adapters participating in shaping. For example, in case of two Fast Ethernet cards this variable shall be set to “`eth0:102400 eth1:102400`”.

The `TOTALRATE` variable specifies the size of the so-called bandwidth pool for each network class being shaped. The bandwidth from the pool can be borrowed by Virtual Private Servers when they need more bandwidth for communicating with hosts from the corresponding network class. It is used to limit the total available outgoing traffic Virtual Private Servers can consume; the next section explains it in more detail. The format of this variable is “`<NIC>:<network_class>:<bandwidth_in_Kbits_per_second>`” and defines the pool size per network class for a given network adapter. Multiple entries for different network classes and adapters shall be separated by spaces. The default value for `TOTALRATE` is “`eth0:1:4096`”, which corresponds to the pool size of 4Mb/s for Network Class 1 on the first Ethernet adapter.

In the Virtuozzo global configuration file, you may also define the `RATE` variable whose value amounts to the number of kilobits per second any VPS is guaranteed to receive for outgoing traffic with a network class on an Ethernet device. The default value of this parameter is “`eth0:1:8`”, which means that any Virtual Private Server is guaranteed to receive the bandwidth of at least 8 Kbits/s for sending data to Class 1 hosts on the first Ethernet device. This bandwidth is not the limit for a VPS (unless the `RATEBOUND` parameter is set to “yes” in the VPS configuration file) – the VPS is able to take the needed bandwidth from the `TOTALRATE` bandwidth pool if it is not used by other VPSs.

After setting up the above variables, start bandwidth management as is illustrated below:

```
# /etc/init.d/vz shaperon
Starting Virtuozzo shaping: Ok
Set shaping on running VPS :
vz WARNING: Can't get tc class for VPS(101).
vz WARNING: Can't access file /var/run/vz_tc_classes. \
Creating new one.
vz WARNING: Can't get tc class for VPS(1).
```

Now you have activated the network bandwidth limits. To turn traffic shaping off temporarily, use the `/etc/init.d/vz shaperoff` command. If you want to disable bandwidth management permanently, set the `TRAFFIC_SHAPING` variable to “no” in the Virtuozzo global configuration file.

VZMC provides a convenient means for turning on/off network bandwidth management on the **Shaping** tab of the **Traffic Accounting and Shaping** window, which you can access by doing the following:

- 1 Select the Traffic Management item under the necessary Node in the left pane of the VZMC window;
- 2 Follow the [Click here to change traffic shaping settings](#) link in the right pane:

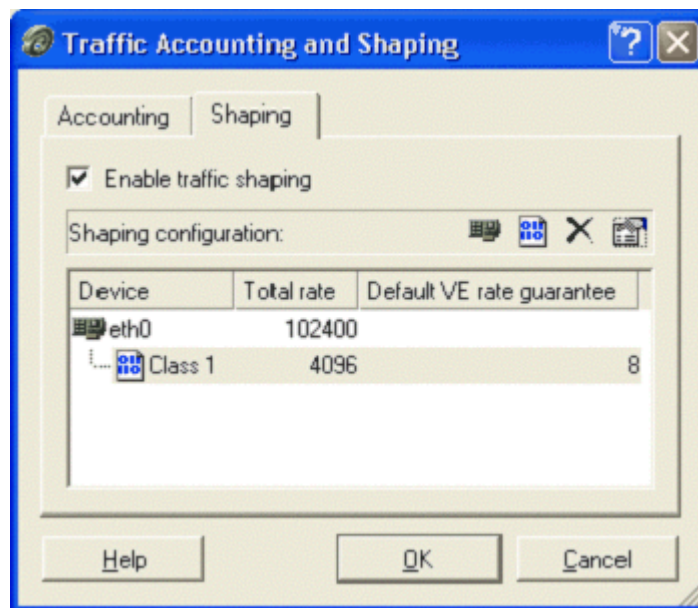


Figure 32: VZMC - Setting Up Traffic Shaping Parameters

In this window you can:

- Enable/disable traffic shaping by selecting/deselecting the corresponding checkbox;
- Add/edit/delete an Ethernet device on the Hardware Node for traffic shaping;
- Add/edit/delete a network class for traffic shaping;
- Set up the `BANDWIDTH` parameter value for each Ethernet device;
- Set up the `TOTALRATE` parameter value for each network class of any Ethernet device;
- Set up the `RATE` parameter value which is the default network bandwidth guarantee for any Virtual Private Server sending data to the given network class over the given Ethernet device.

The traffic shaping settings will take effect immediately on your pressing the OK button in this window.

Configuring Network Bandwidth Management for VPS

The network bandwidth for outgoing traffic a Virtual Private Server receives is controlled by two variables in the Virtual Private Server configuration file (`/etc/sysconfig/vz-scripts/<vpsid>.conf`): `RATE` and `RATEBOUND`.

Note: VPS incoming traffic cannot be controlled in Virtuozzo 3.0.

The `RATE` variable has the same format as `TOTALRATE`: “<NIC>:<network_class>:<bandwidth>”. This variable specifies the guaranteed outgoing traffic rate that the corresponding Virtual Private Server receives. This rate can be specified differently for different network classes and network adapters; use space to separate several rate descriptions.

Bandwidth values are specified in Kbit/s. It is recommended to increase this value in 8Kbit/s chunks and to set it no lower than 8Kbit/s.

The `RATEBOUND` variable specifies whether the network bandwidth available to the Virtual Private Server for outgoing traffic is limited by the bandwidth specified in the `RATE` variable. The possible values of the `RATEBOUND` variable are “yes” and “no”, the default is “no”. In this case the VPS is allowed to take free bandwidth from the `TOTALRATE` pool.

The actual network bandwidth available to the Virtual Private Servers depends on the number of Virtual Private Servers and the total sum of the `RATE` values, and normally does not coincide with the bandwidth specified in their own `RATE` variables. If the `RATEBOUND` variable is set to “yes”, then the VPS bandwidth is limited by the value of the `RATE` variable.

If the Virtual Private Server configuration file does not specify any of these parameters, the values from Virtuozzo global configuration file are taken. By default, Virtuozzo does not set `RATEBOUND`, which corresponds to “no”, and `RATE` is set to “eth0:1:8”.

Virtuozzo network bandwidth management works in the following way. The bandwidth pool for a given network class (configurable through the `TOTALRATE` variable in the Virtuozzo global configuration file) is divided among the Virtual Private Servers transmitting data proportionally to their `RATE` settings. If the total value of the `RATE` variables of all Virtual Private Servers transmitting data does not exceed the `TOTALRATE` value, each Virtual Private Server gets the bandwidth equal or greater than its `RATE` value (unless this Virtual Private Server has the `RATEBOUND` variable set to “yes”). If the total value of the `RATE` variables of all Virtual Private Servers transmitting data exceeds the `TOTALRATE` value, each Virtual Private Server may get less than its `RATE` value.

The example below illustrates the scenario when there are two Virtual Private Servers, 101 and 102, which have `RATEBOUND` set to “no”, and Virtual Private Server 103 has `RATEBOUND` set to “yes”:

```
# grep ^RATE /etc/sysconfig/vz-scripts/101.conf \
/etc/sysconfig/vz-scripts/102.conf
RATE="eth0:1:8"
RATEBOUND="no"
RATE="eth0:1:8"
RATEBOUND="no"
# grep ^RATE /etc/sysconfig/vz-scripts/103.conf
```

```
RATE="eth0:1:64"
RATEBOUND="yes"
```

With the default TOTALRATE of 4096Kbit/s, bandwidth pool will be distributed according to the following table:

VPS 101	VPS 102	VPS 103	Bandwidth consumed by VPSs	
transmits	idle	idle	VPS101: 4096Kbits/s	
idle	idle	transmits	VPS103: 64Kbits/s	
transmits	transmits	idle	VPS101:	2048Kbits/s
			VPS102: 2048Kbits/s	
transmits	idle	transmits	VPS101:	4032Kbits/s
			VPS103: 64Kbits/s	
transmits	transmits	transmits	VPS101:	2016Kbits/s
			VPS102:	2016Kbits/s
			VPS103: 64Kbits/s	

After you have set up Virtual Private Server bandwidth settings, activate your changes as below:

```
# /etc/init.d/vz shaperrestart
Stopping Virtuozzo shaping: Ok
Starting Virtuozzo shaping: Ok
Set shaping on running VPS: Ok
```

This command clears off all existing shaping settings and sets them again using the configuration files of running Virtual Private Servers.

By means of VZMC, you can provide the network bandwidth settings for a particular VPS on the QoS tab of the Virtual Private Server manager window, which you can access by doing the following:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.

- 2 Click the Traffic Shaping tab:

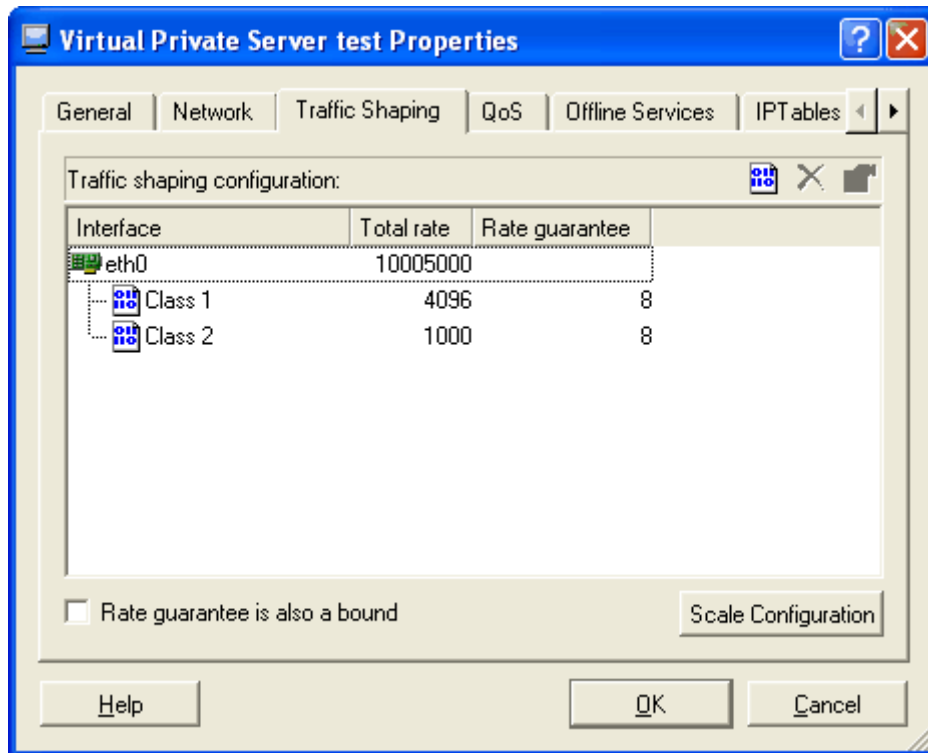


Figure 33: VZMC - Configuring VPS Network Bandwidth

In this window you can:

- 1 Set the value for the RATEBOUND parameter for the given VPS by selecting/deselecting the Rate guarantee is also a bound checkbox;
- 2 Add/edit/delete a network class for traffic shaping for any Ethernet device;
- 3 Set up the RATE guarantee parameter value for the given VPS for any network class of any Ethernet device;

The traffic shaping settings will take effect immediately on your pressing the OK button in this window.

Managing System Parameters

The resources a Virtual Private Server may allocate are defined by the system resource control parameters. These parameters can be subdivided into the following categories: primary, secondary, and auxiliary parameters. The primary parameters are the start point for creating a Virtual Private Server configuration from scratch. The secondary parameters are dependent on the primary ones and are calculated from them according to a set of constraints. The auxiliary parameters help improve fault isolation among applications in one and the same Virtual Private Server and the way applications handle errors and consume resources. They also help enforce administrative policies on Virtual Private Servers by limiting the resources required by an application and preventing the application to run in the Virtual Private Server.

Listed below are all the system resource control parameters. The parameters starting with "num" are measured in integers. The parameters ending in "buf" or "size" are measured in bytes. The parameters containing "pages" in their names are measured in 4096-byte pages. The File column indicates that all the system parameters are defined in the corresponding VPS configuration files (V).

Primary parameters

Parameter	Description	File
avnumproc	The average number of processes and threads.	V
numproc	The maximal number of processes and threads the VPS may create.	V
numtcpsock	The number of TCP sockets (PF_INET family, SOCK_STREAM type). This parameter limits the number of TCP connections and, thus, the number of clients the server application can handle in parallel.	V
numothersock	The number of sockets other than TCP ones. Local (UNIX-domain) sockets are used for communications inside the system. UDP sockets are used, for example, for Domain Name Service (DNS) queries. UDP and other sockets may also be used in some very specialized applications (SNMP agents and others).	V
vmguarpages	The memory allocation guarantee, in pages (one page is 4 Kb). VPS applications are guaranteed to be able to allocate additional memory so long as the amount of memory accounted as <code>privvmpages</code> (see the auxiliary parameters) does not exceed the configured barrier of the <code>vmguarpages</code> parameter. Above the barrier, additional memory allocation is not guaranteed and may fail in case of overall memory shortage.	V

Secondary parameters

Parameter	Description	File
kmemsize	The size of unswappable kernel memory allocated for the internal kernel structures for the processes of a particular VPS.	V
tcpsndbuf	The total size of send buffers for TCP sockets, i.e. the amount of kernel memory allocated for the data sent from an application to a TCP socket, but not acknowledged by the remote side yet.	V

<code>tcprecvbuf</code>	The total size of receive buffers for TCP sockets, i.e. the amount of V kernel memory allocated for the data received from the remote side, but not read by the local application yet.
<code>othersockbuf</code>	The total size of UNIX-domain socket buffers, UDP, and other V datagram protocol send buffers.
<code>dgramrcvbuf</code>	The total size of receive buffers of UDP and other datagram protocols. V
<code>oomguarpages</code>	The out-of-memory guarantee, in pages (one page is 4 Kb). Any VPS V process will not be killed even in case of heavy memory shortage if the current memory consumption (including both physical memory and swap) does not reach the <code>oomguarpages</code> barrier.

Auxiliary parameters

Parameter	Description	File
<code>lockedpages</code>	The memory not allowed to be swapped out (locked with the <code>V mlock()</code> system call), in pages.	V
<code>shmpages</code>	The total size of shared memory (including IPC, shared anonymous V mappings and <code>tmpfs</code> objects) allocated by the processes of a particular VPS, in pages.	V
<code>privvmpages</code>	The size of private (or potentially private) memory allocated by an V application. The memory that is always shared among different applications is not included in this resource parameter.	V
<code>numfile</code>	The number of files opened by all VPS processes.	V
<code>numflock</code>	The number of file locks created by all VPS processes.	V
<code>numpty</code>	The number of pseudo-terminals, such as an <code>ssh</code> session, the <code>screen</code> V or <code>xterm</code> applications, etc.	V
<code>numsiginfo</code>	The number of <code>siginfo</code> structures (essentially, this parameter limits V the size of the signal delivery queue).	V
<code>dcachesize</code>	The total size of <code>dentry</code> and <code>inode</code> structures locked in the V memory.	V
<code>physpages</code>	The total size of RAM used by the VPS processes. This is an V accounting-only parameter currently. It shows the usage of RAM by the VPS. For the memory pages used by several different VPSs (mappings of shared libraries, for example), only the corresponding fraction of a page is charged to each VPS. The sum of the <code>physpages</code> usage for all VPSs corresponds to the total number of pages used in the system by all the accounted users.	V
<code>numiptent</code>	The number of IP packet filtering entries.	V

You can edit any of these parameters in the `/etc/sysconfig/vz-scripts/<VPSID>.conf` file of the corresponding VPS by means of your favorite text editor (for example, `vi` or `emacs`), or by running the `vzctl set` command. For example:

```
# vzctl set 101 --kmemsize 2211840:2359296 --save
Saved parameters for VPS 101
```

In VZMC, to view and/or change any of these parameters for a particular Virtual Private Server, do the following:

- 1 Click **Virtual Private Servers** in the VZMC left pane, right-click the needed VPS in the right pane, and choose **Properties**.

- 2** Click the QoS tab and expand the corresponding group: Primary UBC parameters, Secondary UBC parameters, or Auxiliary UBC parameters.
- 3** Double-click the necessary parameter, and, if necessary, enter the right value for the given VPS.
- 4** Press OK.

Note: Detailed information on managing system parameters is provided in the *Virtuozzo Management of System Resources Administrator's Guide*.

Monitoring System Resources Consumption

It is possible to check the system resource control parameters statistics from within a Virtual Private Server. The primary use of these statistics is to understand what particular resource has limits preventing an application to start. Moreover, these statistics report the current and maximal resources consumption for the running Virtual Private Server. This information can be obtained from the `/proc/user_beancounters` file.

The output below illustrates a typical session:

```
# vzctl exec 101 cat /proc/user_beancounters
Version: 3.0.0
uid resource          held      maxheld   barrier   limit      failcnt
101: kmemsize          583072    671083    2211840   2359296     0
      lockedpages      0          0          32         32          0
      privmmpages      2695      2987      22528     24576        0
      shmpages         395       395       1024      1024         0
      numproc          13        16         65         65          0
      physpages        1784      2066         0  2147483647    0
      vmguarpages       0          0        6144  2147483647    0
      oomguarpages     1784      2066      6144  2147483647    0
      numtcpsock        4          4          80         80          0
      numflock          1          2         100        110          0
      numpty            0          0          16         16          0
      numsiginfo        0          6         256        256          0
      tcpsndbuf         0          0      319488    524288        0
      tcprcvbuf         0          0      319488    524288        0
      othersockbuf     2224      9376     122880    327680        0
      dgramrcvbuf       0          0      65536     65536         0
      numothersock      4          9          80         80          0
      dcachesize       57684     69597    491520    524288        0
      numfile           190        218       1280      1280         0
      numiptent         4          4         128        128          0
```

The `failcnt` column displays the number of unsuccessful attempts to allocate a particular resource. If this value increases after an application fails to start, then the corresponding resource limit is in effect lower than is needed by the application.

The `held` column displays the current resource usage, and the `maxheld` column – the maximal value of the resource consumption for the last accounting period. The meaning of the `barrier` and `limit` columns depends on the parameter and is explained in the *Virtuozzo Management of System Resources Guide*.

Inside a VPS, the `/proc/user_beancounters` file displays the information on the given VPS only, whereas from the Hardware Node this file displays the information on all the VPSs.

To check the UBC usage for a HN (summary for all running VPSs), you can use the following scripts:

(for any resource accounted in pages)

```
# for res in lockedpages totvmmpages ipcshmpages anonshpages rsspages; \
do echo;echo "$res usage for all VEs, in MB:";cat \
/proc/user_beancounters |grep $res|awk 'BEGIN{ cur=max=lim=0; } \
{ cur+=$2; max+=$3;lim+=$5 } END {print "held:",cur*4/1024, "max:", \
max*4/1024, "limit:", lim*4/1024}'; done
```

(for kmemsize and other resources accounted in bytes)

```
# for res in tcpsendbuf tcprcvbuf unixsockbuf sockrcvbuf kmemsize; \
do echo;echo "$res usage for all VEs, in MB:";cat \
/proc/user_beancounters |grep $res|sed "s/[[digit:]]\+://g" \
|awk 'BEGIN{ cur=max=lim=0; } { cur+=$2; max+=$3;lim+=$5 } \
END {print "held:",cur/1024/1024, "max:", max/1024/1024, \
"limit:", lim/1024/1024}'; done
```

Monitoring Memory Consumption

You can monitor a number of memory parameters for the whole Hardware Node and for particular Virtual Private Servers with the help of the `vzmemcheck` utility. For example:

```
# vzmemcheck -v
Output values in %
veid      LowMem  LowMem      RAM MemSwap MemSwap      Alloc  Alloc  Alloc
          util   commit    util   util   commit    util   commit limit
101       0.19    1.93     1.23    0.34    1.38     0.42    1.38   4.94
1         0.27    8.69     1.94    0.49    7.19     1.59    2.05  56.54
-----
Summary:  0.46   10.62     3.17    0.83    8.57     2.02    3.43  61.48
```

The `-v` option is used to display the memory information for each Virtual Private Server and not for the Hardware Node in general. It is also possible to show the absolute values in Megabytes by using the `-A` switch. The monitored parameters are (from left to right in the output above) low memory utilization, low memory commitment, RAM utilization, memory+swap utilization, memory+swap commitment, allocated memory utilization, allocated memory commitment, allocated memory limit.

To understand these parameters, let us first draw the distinction between utilization and commitment levels. *Utilization* level is the amount of resources consumed by VPSs at the given time. In general, low utilization values mean that the system is under-utilized. Often, it means that the system is capable of supporting more Virtual Private Servers if the existing VPSs continue to maintain the same load and resource consumption level. High utilization values (in general, more than 1, or 100%) mean that the system is overloaded and the service level of the Virtual Private Servers is degraded. *Commitment* level shows how much resources are “promised” to the existing Virtual Private Servers. Low commitment levels mean that the system is capable of supporting more Virtual Private Servers. Commitment levels more than 1 mean that the Virtual Private Servers are promised more resources than the system has, and the system is said to be *overcommitted*. If the system runs a lot of VPSs, it is usually acceptable to have some overcommitment because it is unlikely that all Virtual Private Servers will request resources at one and the same time. However, very high commitment levels will cause VPSs to fail to allocate and use the resources promised to them and may hurt system stability.

There follows an overview of resources checked up by the `vzmemcheck` utility. Their complete description is provided in the *Virtuozzo Administrator’s Guide to Management of System Resources* shipped with Virtuozzo 3.0.

The *low memory* is the most important RAM area representing the part of memory residing at lower addresses and directly accessible by the kernel. In Virtuozzo 2.5.1 release, the size of the “low” memory area is limited to 832MB. In Virtuozzo 2.5.1 Service Pack 1 and later releases, the “low” memory area size is limited to 832MB in the UP (uniprocessor) version of the kernel and to 3.6GB in the SMP and Enterprise versions of the kernel. If the total size of the computer RAM is less than the limit (832MB or 3.6GB, respectively), then the actual size of the “low” memory area is equal to the total memory size.

The union of *RAM and swap* space is the main computer resource determining the amount of memory available to applications. If the total size of memory used by applications exceeds the RAM size, the Linux kernel moves some data to swap and loads it back when the application needs it. More frequently used data tends to stay in RAM, less frequently used data spends more time in swap. Swap-in and swap-out activity reduces the system performance to some extent. However, if this activity is not excessive, the performance decrease is not very noticeable. On the other hand, the benefits of using swap space are quite big, allowing to increase the number of Virtual Private Servers in the system by 2 times. Swap space is essential for handling system load bursts. A system with enough swap space just slows down at high load bursts, whereas a system without swap space reacts to high load bursts by refusing memory allocations (causing applications to refuse to accept clients or terminate) and directly killing some applications. Additionally, the presence of swap space helps the system better balance memory and move data between the low memory area and the rest of the RAM.

Allocated memory is a more “virtual” system resource than the RAM or RAM plus swap space. Applications may allocate memory but start to use it only later, and only then will the amount of free physical memory really decrease. The sum of the sizes of memory allocated in all Virtual Private Servers is only the estimation of how much physical memory will be used if all applications claim the allocated memory. The memory available for allocation can be not only used (the *Alloc util* column) or promised (the *Alloc commit* column), but also limited (applications will not be able to allocate more resources than is indicated in the *Alloc limit* column).

Managing VPS Resources Configuration

Any VPS is configured by means of its own configuration file. You can manage your VPS configurations in a number of ways:

- 1 Using configuration sample files shipped with Virtuozzo 3.0. These files are used when a new Virtual Private Server is being created (for details, see the **Creating and Configuring New Virtual Private Server** section (on page 31)). They are stored in the same directory as VPS configuration files (*/etc/sysconfig/vz-scripts/*) and have the *ve-<name>.conf-sample* mask. Currently, five configuration sample files are provided:
 - *vps.basic* – to be used for common VPSs;
 - *vps.confixx* – to be used for creating VPSs that are to use the Confixx control panels.
 - *vps.cpanel* - to be used for creating VPSs where the CPANEL application is to be installed;
 - *vps.plesk7.rh9* - to be used for creating VPSs based on the Red Hat 9 template with the Plesk 7 control panel;
 - *vps.256Mb* - to be used for creating VPSs with 256 Mb of main memory;
 - *vps.512Mb* - to be used for creating VPSs with 512 Mb of main memory;
 - *vps.1024Mb* - to be used for creating VPSs with 1024 Mb of main memory;
 - *vps.2048Mb* - to be used for creating VPSs with 2048 Mb of main memory.

Note: Configuration sample files cannot contain spaces in their names.

Any sample configuration file may also be applied to a Virtual Private Server after it has been created. You would do this if, for example, you want to upgrade or downgrade the overall resources configuration of a particular VPS:

```
# vzctl set 101 --applyconfig vps.basic --save
```

This command applies all the parameters from the `ve-vps.basic.conf-sample` file to the given VPS, except for the `OSTEMPLATE`, `TEMPLATES`, `VE_ROOT`, `VE_PRIVATE` parameters, should they exist in the sample configuration file.

- 2 Using Virtuozzo specialized utilities for preparing configuration files in their entirety. The tasks these utilities perform are described in the following subsections of this section.
- 3 The direct creating and editing of the corresponding configuration file (`/etc/sysconfig/vz-scripts/<VPS_ID>.conf`). This can be performed either with the help of any text editor or thru Virtuozzo Management Console. The instructions on how to edit VPS configuration files directly are provided in the four preceding sections. In this case you have to edit all the configuration parameters separately, one by one.

Splitting Hardware Node Into Equal Pieces

It is possible to create a Virtual Private Server configuration roughly representing a given fraction of the Hardware Node. If you want to create such a configuration that up to 20 fully loaded Virtual Private Servers would be able to be simultaneously running on the given Hardware Node, you can do it as is illustrated below:

```
# cd /etc/sysconfig/vz-scripts/
# vzsplit -n 20 -f vps.mytest
Config /etc/sysconfig/vz-scripts/ve-vps.mytest.conf-sample was created
# vzcfgvalidate ve-vps.mytest.conf-sample
Recommendation: kmemsize.lim-kmemsize.bar should be > 253952 \
(currently, 126391)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 93622)
```

Note that the configuration produced depends on the given Hardware Node resources. Therefore, it is important to validate the resulted configuration file before trying to use it, which is done with the help of the `vzcfgvalidate` utility.

The number of Virtual Private Servers you can run on the Hardware Node is actually several times greater than the value specified in the command line because Virtual Private Servers normally do not consume all the resources that are guaranteed to them. To illustrate this idea, let us look at the Virtual Private Server created from the configuration produced above:

```
# vzctl create 101 --pkgset redhat-7.3 --config vps.mytest
Creating VE private area
Getting private area from cache: /vz/private/pool/redhat-7.3/4
vzquota : (error) Can't open quota file for id 101, maybe you need \
to reinitialize quota: No such file or directory
Initializing quota ...
VE private area was created
# vzctl set 101 --ipadd 192.168.1.101 --save
Saved parameters for VE 101
# vzctl start 101
Starting VE ...
VE is mounted
Adding IP address(es): 192.168.1.101
VE start in progress...
# vzcalc 101
```

Resource	Current(%)	Promised(%)	Max(%)
Memory	0.53	1.90	6.44

As is seen, if Virtual Private Servers use all the resources guaranteed to them, then around 20 VPSs can be simultaneously running. However, taking into account the **Promised** column output, it is safe to run 40-50 such Virtual Private Servers on this Hardware Node.

There is a possibility to create a suchlike configuration sample file using VZMC:

- 1 Select the **VPS Sample Configurations** option in the Hardware Node main tree;
- 2 Choose **"Slice" Hardware Node** on the **Action** menu;
- 3 Follow the instructions of the wizard. The resulting file will be saved in the default directory (`/etc/sysconfig/vz-scripts/`) with the name you have specified.

Scaling Virtual Private Server Configuration

Any configuration or configuration sample file can prove insufficient for your needs. You might have an application, which does not fit into existing configurations. The easiest way of producing a Virtual Private Server configuration is to scale an existing one.

Scaling produces a “heavier” or “lighter” configuration in comparison with an existing one. All the parameters of the existing configuration are multiplied by a given number. A heavier configuration is produced with a factor greater than 1, and a lighter one – with a factor between 0 and 1.

The session below shows how to produce a configuration sample 50% heavier than the `vps.basic` configuration shipped with Virtuozzo:

```
# cd /etc/sysconfig/vz-scripts/
# vzcfgscale -a 1.5 -o ve-vps.improved.conf-sample \
ve-vps.basic.conf-sample
# vzcfgvalidate ve-vps.improved.conf-sample
Recommendation: kmemsize.lim-kmemsize.bar should be > 245760 \
(currently, 221184)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 98304)
Validation completed: success
```

Now `vps.improved` can be used in the `vzctl create` command for creating new Virtual Private Servers.

It is possible to use the same technique for scaling configurations of the existing Virtual Private Servers. Please note that the output file cannot be the same as the file being scaled. You have to save the scaling results into an intermediate file.

In VZMC, on the contrary, the scaling results are not written into a new file. If you scale the configuration of a Virtual Private Server, its configuration file is changed without saving the original file. If you scale a configuration sample file, it is correspondingly modified. That is why, it is recommended to create a copy of the configuration sample file you are going to scale before scaling it.

To scale an existing configuration using VZMC, do the following:

- 1 Select the **VPS Sample Configurations** or **Virtual Private Servers** option in the Hardware Node main tree;
- 2 Right-click the sample configuration file or the VPS configuration file of which you are going to scale and select **Properties**;
- 3 Go to the **QoS** tab and press the **Scale Configuration** button;
- 4 Determine whether you want to enhance or attenuate the current configuration and specify the factor;
- 5 You may choose what group(s) of parameters will be scaled under the **Apply scaling to group**;
- 6 You are strongly encouraged to validate the resulting configuration with the help of the **Validate** button before pressing **OK**;
- 7 Press **OK** to save the changes;

- 8** Go to the **Traffic Shaping** tab in the **Configuration Properties** window and press the **Scale Configuration** button;
- 9** Determine whether you want to enhance or attenuate the current configuration and specify the factor;
- 10** Press **OK** to save the changes.

Validating Virtual Private Server Configuration

The system resource control parameters have complex interdependencies. Violation of these interdependencies can be catastrophic for the Virtual Private Server. In order to ensure that a Virtual Private Server does not break them, it is important to validate the VPS configuration file before creating VPSs on its basis.

The typical validation scenario is shown below:

```
# vzcfgvalidate /etc/sysconfig/vz-scripts/101.conf
Error: kmemsize.bar should be > 1835008 (currently, 25000)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
Recommendation: othersockbuf.bar should be > 132096 \
(currently, 122880)
# vzctl set 101 --kmemsize 2211840:2359296 --save
Saved parameters for VE 101
# vzcfgvalidate /etc/sysconfig/vz-scripts/101.conf
Recommendation: kmemsize.lim-kmemsize.bar should be > 163840 \
(currently, 147456)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
Recommendation: othersockbuf.bar should be > 132096 \
(currently, 122880)
Validation completed: success
```

The utility checks constraints on the resource management parameters and displays all the constraint violations found. There can be three levels of violation severity:

Recommendation	This is a suggestion, which is not critical for Virtual Private Server or Hardware Node operations. The configuration is valid in general; however, if the system has enough memory, it is better to increase the settings as advised.
Warning	A constraint is not satisfied, and the configuration is invalid. The Virtual Private Server applications may not have optimal performance or may fail in an ungraceful way.
Error	An important constraint is not satisfied, and the configuration is invalid. The Virtual Private Server applications have increased chances to fail unexpectedly, to be terminated, or to hang.

In the scenario above, the first run of the `vzcfgvalidate` utility found a critical error for the `kmemsize` parameter value. After setting reasonable values for `kmemsize`, the resulting configuration produced only recommendations, and the Virtual Private Server can be safely run with this configuration.

You can also validate any configuration sample file the given Hardware Node has by means of VZMC. To this effect, do the following:

- 1 Click **VPS Sample Configurations** in the VZMC left pane, right-click the needed sample configuration file in the right pane, and choose **Properties**.
- 2 Click the **QoS** tab and press the **Verify Configuration** button.

3 A window appears informing you of the results. For example:

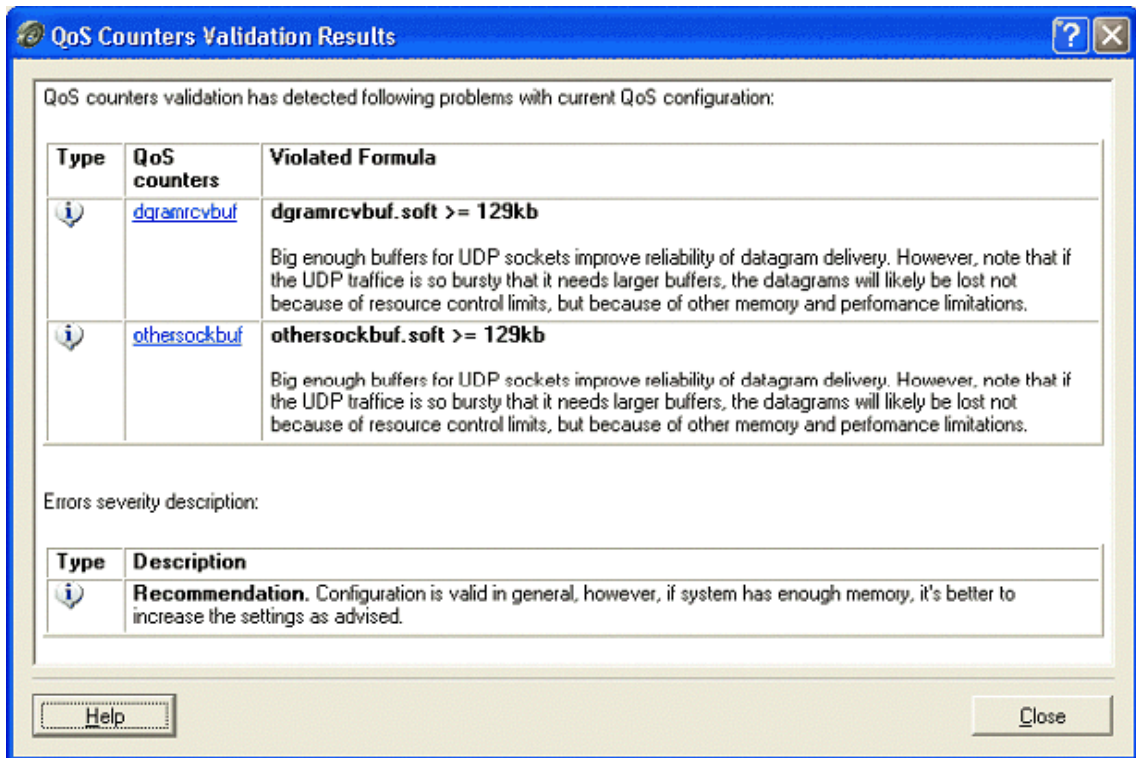


Figure 34: VZMC - Validating VPS Configuration

In this example, only the recommendations are shown, therefore, the configuration is in general valid.

CHAPTER 7

Real-Time Monitoring in Virtuozzo

In This Chapter

Monitoring Resources in Text Console.....	154
Monitoring Resources in VZMC	156
Subscribing for Alerts	165

Monitoring Resources in Text Console

Virtuozzo 3.0 includes quite a number of means to monitor the Hardware Node and Virtual Private Servers resources. One of Virtuozzo's most powerful features is the ability to monitor resources in real time. To this effect, you may run the `vzstat` utility on the Hardware Node, for example, with the following options:

```
# vzstat -d 5 -v
1:01pm, up 1 day, 0:53, 1 user, load average: 0.19, 0.57, 0.54
VNum 3, procs 9: running 1, sleeping 8, unint 0, zombie 0, stopped 0
CPU [ OK ]: VEs 0%, VE0 0%, user 0%, sys 0%, idle 99%, lat(ms) 0/0
Mem [ OK ]: total 493MB, free 17MB/0MB (low/high), lat(ms) 0/0
ZONE0 (DMA): size 16MB, act 3MB, inact 3MB, free 2MB (0/0/0)
fragm 127*1 9*2 1*4 1*8 1*16 1*32 0*64 0*128 0*256 1*512
ZONE1 (Normal): size 487MB, act 243MB, inact 99MB, free 14MB (0/1/2)
fragm 2592*1 283*2 10*4 0*8 3*16 1*32 0*64 1*128 1*256 0*512
Mem lat (ms): A0 0, K0 0, U0 0, K1 0, U1 0
Slab pages: 165MB/165MB (ino 113MB, de 23MB, bh 26MB, pb 0MB)
Swap [ OK ]: tot 1520MB, free 1520MB, in 0.000MB/s, out 0.000MB/s
Swap lat: si 31832, 0/0 ms, so 423, 0/0 ms, 0/0 cpu ms
Swap cache: add 10546, del 6610, find 31750/31832
Net [ OK ]: tot in 0.002MB/s 14pkt/s, out 0.000MB/s 7pkt/s
lo: in 0.000MB/s 7pkt/s, out 0.000MB/s 7pkt/s
eth0: in 0.001MB/s 7pkt/s, out 0.000MB/s 0pkt/s
Disks [ OK ]: in 0.000MB/s, out 0.007MB/s
root(/) free: 923MB(31%), 331521ino(86%)
hda1(/boot) free: 48MB(83%), 15387ino(99%)
hda7(/vz) free: 12105MB(92%), 15129402ino(99%)

VEID ST %VM %KM PROC CPU SOCK FCNT MLAT IP
1 OK 7.0/204 0.1/2.4 0/21/256 0.0/50 32/708 0 0 10.0.186.1
101 OK 1.6/18 0.1/0.5 0/8/65 0.0/50 8/160 0 0 10.0.186.101
```

This screen will be updated with the time interval equal to the value specified after the `-d` (delay) option measured in seconds. In the session above, the statistics displayed will be renewed every five seconds. If the `-d` option is not specified, the default interval equals 1 second.

As you can see, the utility provides real-time information on the number of VPSs and processes (in each and every state) on the Hardware Node, as well as on all the main resources subsystems pertaining both to the Hardware Node and to its Virtual Private Servers – the disk, network, CPU, and memory subsystems. You may want to shrink the output of the utility by specifying the `-b` (brief) option instead of the `-v` (verbose) one, or to do without any options to use the “normal” mode of displaying.

The following information is displayed per each VPS:

Column Name	Description
VEID	Virtual Private Server ID

ST	VPS status. If there are no failed counters and the latency values are normal, the status is “OK”. Otherwise, it is displayed in red as “!!”. You can sort VPSs by their status to see the problem VPSs first.
%VM	Virtual memory usage (in per cent to the total memory), corresponding to the <code>privvmpages</code> parameter. The first number is how much <code>privvmpages</code> are being held, and the second one is the <code>privvmpages</code> barrier.
%KM	Kernel memory usage (in per cent to the normal zone size), corresponding to the <code>kmemsize</code> parameter. The first number is how much <code>kmemsize</code> is being used, and the second one is the <code>kmemsize</code> barrier.
PROC	Running/total/maximal processes number. The maximal number of processes represents the VPS barrier. You can sort the VPSs by the number of running or total processes.
CPU	CPU usage in per cent to all available CPUs. The first number is how much of the CPU power is being used by the VPS, and the second one is its guaranteed share judging by the <code>cpuunits</code> parameter. Note that the actual CPU usage may be higher than the guaranteed one.
SOCK	Sockets usage, corresponding to the sum of the <code>numtcpsock</code> and <code>numothersock</code> parameters. The first number is how many sockets are opened, the second one is the sockets barrier.
FCNT	The number of VPS failed counters for all the resource parameters. In the standard mode of displaying, this number represents the increase of failed counters since the previous screen update, whereas in the average mode of displaying, it represents an absolute failed counters sum for the given VPS.
MLAT	Maximal scheduling latency for the VPS, in ms. This parameter shows the maximal scheduling latency inside the given VPS, i.e. for how long (at the utmost) a process inside the VPS awaits for the CPU.
IP/HOSTNAME	The IP address or the hostname of the given VPS. You may switch between them by pressing the <code>e</code> key on the keyboard while <code>vzstat</code> is running.

The `%VM`, `%KM`, `CPU`, and `SOCK` columns provide two values per column separated by a slash for each Virtual Private Server. The first value indicates the real usage of the corresponding parameter by the VPS, and the second one – the maximal value allowed for the Virtual Private Server. The `PROC` column shows the number of processes in the corresponding Virtual Private Server in the following format: running/total/maximal number of processes.

The great thing about the `vzstat` utility is its interactivity. You can set the time interval, manage the mode of displaying, sort the Virtual Private Servers by a number of parameters, and all this on-the-fly. For example:

- 1** While `vzstat` is running, press `t` on the keyboard, enter the new timeout (say, 180), and press ENTER;
- 2** Press `b` to switch to the brief details level;
- 3** Press `w` to toggle the display of the swap information on the screen;
- 4** Press `o`, and then `r` to sort the displayed VPSs by the number of running processes.

Now your screen must look something like the following:

```
1:28pm, up 1 day, 1:20, 1 user, load average: 0.00, 0.00, 0.00
VNum 3, procs 9: running 1, sleeping 8, unint 0, zombie 0, stopped 0
CPU [ OK ]: VEs 0%, VE0 0%, user 0%, sys 0%, idle 99%, lat(ms) 0/0
Mem [ OK ]: total 502MB, free 8MB/0MB (low/high), lat(ms) 0/0
Net [ OK ]: tot in 0.000MB/s 2pkt/s, out 0.000MB/s 0pkt/s
Disks [ OK ]: in 0.000MB/s, out 0.000MB/s
```

VEID	ST	%VM	%KM	PROC	CPU	SOCK	FCNT	MLAT	IP
1	OK	7.0/204	0.1/2.4	0/20/256	0.0/50	32/708	0	0	10.0.186.186
101	OK	1.6/18	0.1/0.5	0/8/65	0.0/50	8/160	0	0	10.0.186.1

The `vzstat` utility has a configuration file where you can set the values of different parameters indicating the warning and/or the error levels for them. If a parameter hits the warning level, it will be displayed in yellow by the utility, if it hits the error level – in red. Moreover, if a parameter has hit the error level, the **FAIL** warning is displayed instead of **OK** after the name of the corresponding subsystem (CPU, Memory, Swap, Net, or Disks). Thus, for example, if you see `Swap [FAIL]` on the screen, it means that one or more of the Hardware Node swap-related parameters (the total size of swap memory used, the swap in/out activity, etc.) has hit the error level. The offending parameter(s) will be displayed in red.

Please consult the [Virtuozzo Reference Guide](#) for a complete list of command line options, interactive keys, and configuration file parameters of the `vzstat` utility.

Monitoring Resources in VZMC

You can exploit the **Monitor** feature of VZMC for monitoring resources. This feature provides either the whole Hardware Node resources monitoring or the monitoring of resources consumption by a single VPS, depending on whether you use the VZMC main window or a particular VPS manager window. To open the latter, it is enough to double-click the necessary Virtual Private Server in the VPS table in the right pane of the VZMC main window. The principles of working with these two kinds of monitors are essentially the same (only the set of the parameters that can be displayed is slightly different); therefore, they can be described together.

To access the VZMC **Monitor** feature, you may either select the **Monitor** item in the left pane of the window you are working with, or open a separate **Monitor** console by clicking the **Monitor Hardware Node** button in the VZMC main window toolbar and register the necessary Hardware Node(s) and/or Virtual Private Server(s). In general, it does not make any difference whether a separate **Monitor** console is opened for a Hardware Node or for one of the VPSs. As soon as the **Monitor** console is opened, you can add Hardware Node(s) and/or VPS(s). Moreover, several **Monitor** consoles can be opened for several Hardware Nodes and VPSs in each console.

The windows in the **Monitor** console are identical to those in the **Monitor** item, so you may choose either. To simplify things, we speak only of the **Monitor** item in all the instructions below in this section.

Using Charts Representation

The charts section of VZMC lets you display quite a number of charts for monitoring various kinds of resources on a single grid. It offers means for better visualization of charts, like assigning colors and line styles to all the elements of the grid and charts or choosing a peculiar representation scale for each chart. You can save and load a set of counters you would usually monitor thus avoiding the necessity of adding the counters one by one each time you start VZMC. And, last but not least, there is a possibility to replay the charts for any specified period of time by using logs.

The sequence of your actions may be the following:

- 1 To display the chart, expand the **Monitor** item in the window you are working with (either the VZMC main window or a VPS manager window), and click **Charts** to see the monitor grid in the right pane.
- 2 Right-click in any place of the grid and select **Add Counters**, or press the **Add Counters** button on the **Charts** toolbar.
- 3 In the **Add Monitoring Counters** dialog window, select the set of counters from which you want to add one(s) by selecting the desired group on the **Counter type** drop-down menu.
- 4 Select the needed counter(s) and click **Add**. You may use the Ctrl and Shift keys to add a number of counters from a group. When you select a certain counter with your mouse, the counter description is provided in the lower part of the **Add Monitoring Counters** dialog window. For example:

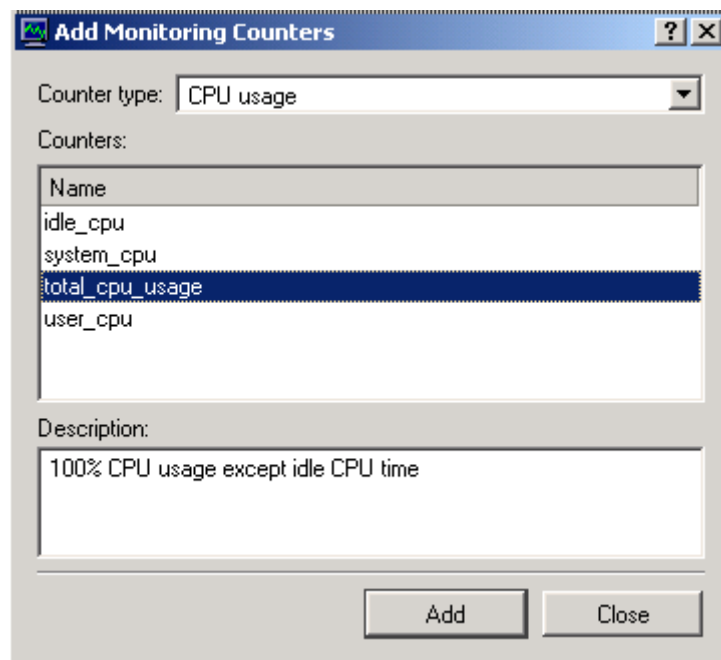


Figure 35: VZMC - Adding Monitoring Counters

- 5 Click **Close** after you have added all the desired counters.

Now that you have a number of counters on the grid, you can see a red line indicating the current moment of time moving from left to right as time passes and new values of monitored parameters appear on the grid. Now it's time to customize your view and learn the other opportunities. You may want to perform the following tasks:

- Adjust the periodicity of refreshing the information on the grid;
- Adjust the representation scale for each counter;
- Adjust colors and line styles for the visual elements;
- Highlight a certain counter;
- Save the current configuration of counters to be able to open it at any moment of time;
- Use the grid to replay some past real-time information about a set of parameters.

Adjusting Periodicity of Refreshing Information

To set the time interval at which the information is refreshed for all the charts, do the following:

- 1 Right-click on the **Monitor** item in the Hardware Node or Virtual Private Server main tree and choose **Properties**;
- 2 In the Monitor Properties window, select the **Charts** line:

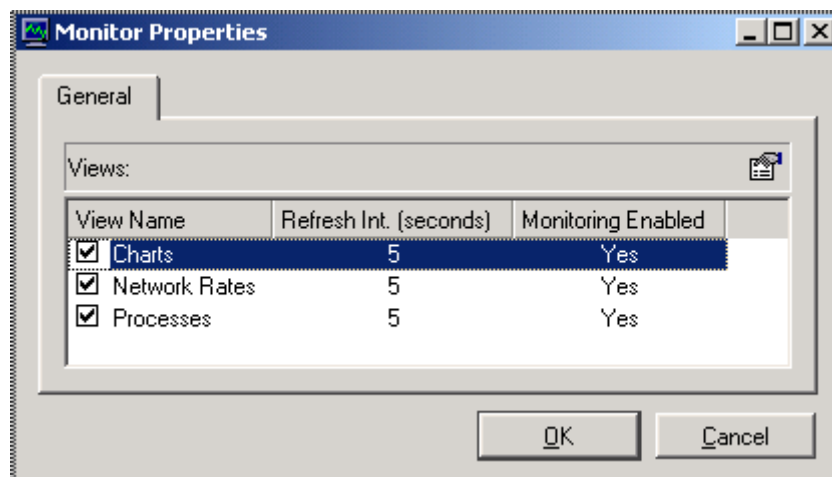


Figure 36: VZMC - Editing Monitor Update Interval

- 3 Press the **Set Refresh Interval** button in the right corner of the **Monitor Properties** window;
- 4 Enter the new time interval (in seconds) and press **OK**.

Adjusting Representation Scale

The value of any counter on the grid may vary from 0 to 100. These numbers are marked on the left of the grid. But the “weight” of these numbers is different for each counter. It is difficult to use one and the same scale, for example, for memory usage which may amount to hundreds of thousands of KBs and for CPU usage in percent. You can adjust the scale for each parameter separately for their better visualization on the grid:

- 1 Click on the name of the corresponding counter in the table of displayed counters below the grid:

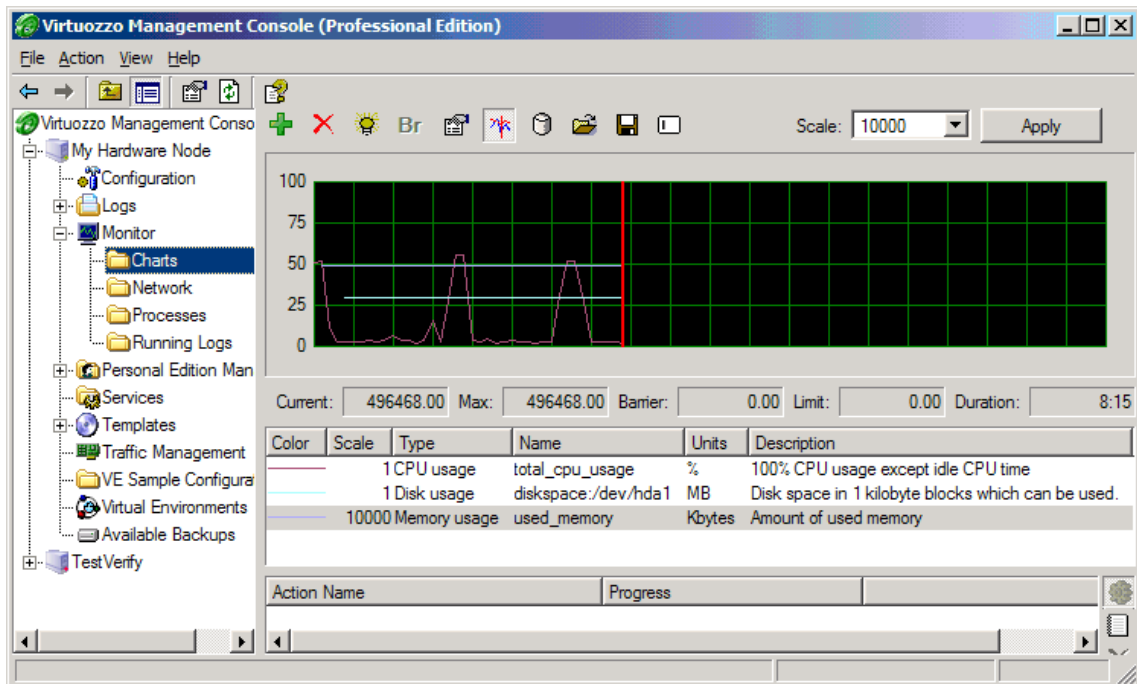


Figure 37: VZMC - Adjusting Charts Scale

- 2 Select the necessary scale on the **Scale** drop-down menu on top of the grid and click **Apply**.

Adjusting Colors and Styles

You can define the way this or that counter is displayed on the grid:

- 1 Right-click on the name of the corresponding counter in the table of displayed counters below the grid and choose **Properties**:

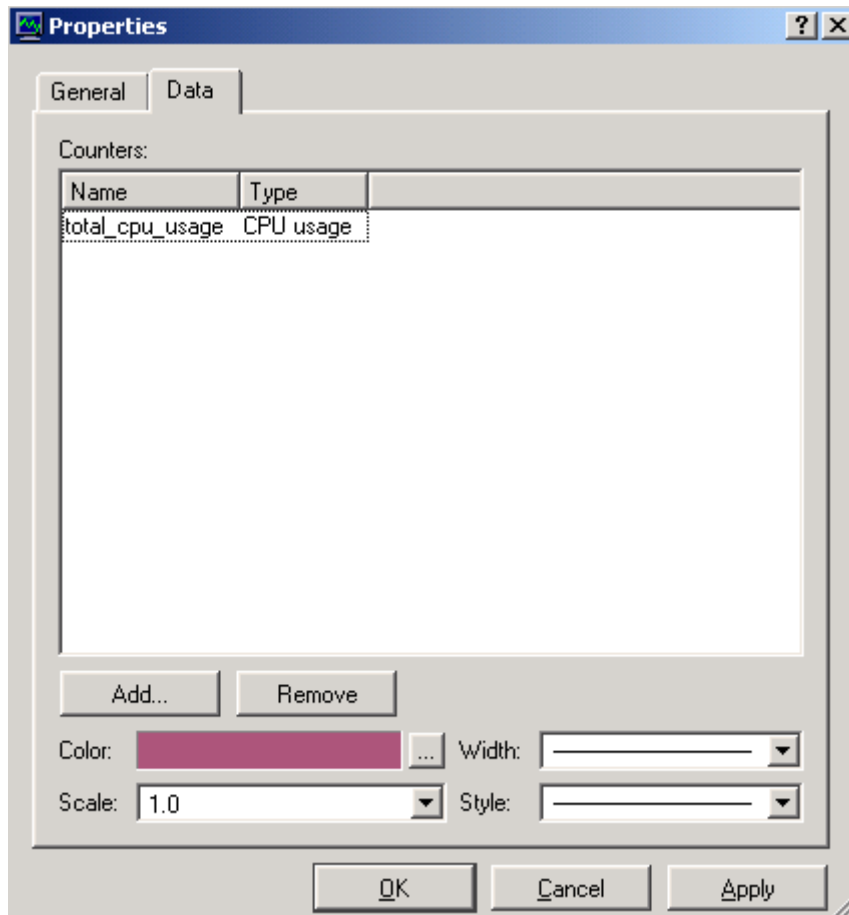


Figure 38: VZMC - Adjusting Charts View

- 2 In the corresponding boxes, adjust the color of the counter line, its width and style as desired;
- 3 Click the **General** tab and adjust the view of the grid elements. The options on that tab are self-explaining.
- 4 Click **OK**.

Highlighting Counter

In case there are many counters being simultaneously displayed on the grid, it might be difficult to quickly single out the needed one. VZMC provides a means for highlighting any one of the counters at a time:

- 1 Click on the name of the corresponding counter in the table of displayed counters below the grid;
- 2 Press the **Highlight Counter** button on the toolbar.

The selected counter will be highlighted on the grid with a broad white line.

Saving Counters Configuration

You can save the information about the current set of counters in the VZMC configuration file to call this information the next time it is needed sparing the labor of adding the counters one by one again. Only one set of counters can thus be saved. Just press the **Save Counters** button on the toolbar after all the desired counters have been added. When you alter the counters configuration (for example, when you restart VZMC, all the counters are erased) and wish to restore the saved configuration, press the **Load Counters** button. The saved set of counters will be loaded from the configuration file.

Replaying Information From Logs

The function of replaying the resources consumption information over a specified time span in the past is ensured by the background logging of all the parameters in Virtuozzo. The default periodicity of refreshing the resources consumption information in the logs is set to be 1 (one) hour. You may have the logs collect the resources consumption information more frequently by "accelerating" the necessary logs with the help of the **Log Setup** item under the **Monitor** item. For example:

- 1 Click **Log Setup** under the **Monitor** item.
- 2 In the provided table, double-click the necessary log group, or right-click it and select **Change Logging Period** on the context menu.

- 3 In the **Change Logging Period** window, set the update period for the given group of logs and the time they are to be stopped. Besides, you can specify the time when the logging period will be set to the default value (1 hour) again by selecting the corresponding radio button:

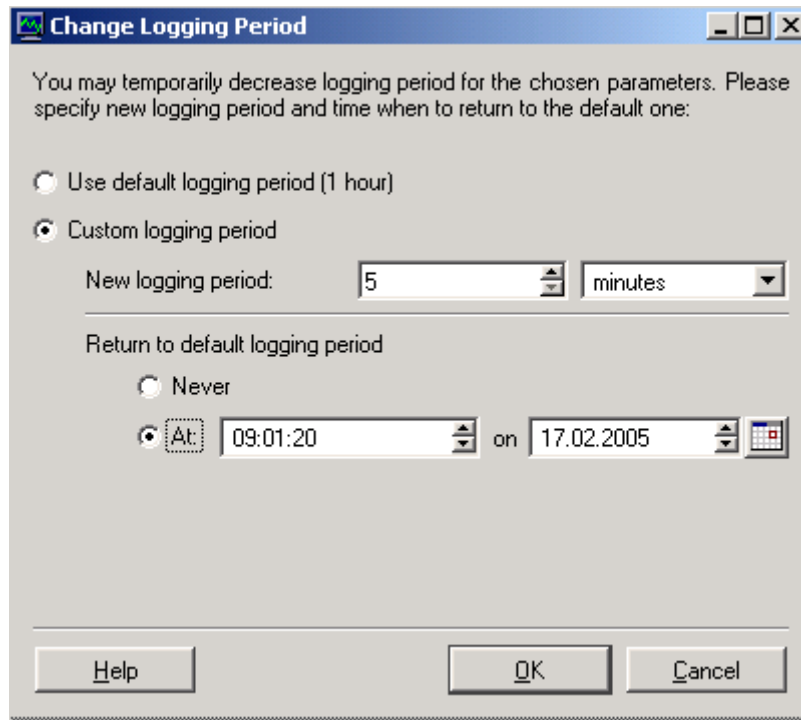


Figure 39: VZMC - Accelerating Logs

- 4 Press **OK** to close the window.

The replaying proper of logs is performed using the same grid of the **Charts** function as for real-time monitoring. The counters are also displayed and configured in the same way as for real-time monitoring. The principal difference is that when replaying the counters, the information for the charts is taken from the logs (both the default logs and the logs accelerated in the **Log Setup** section are used), and not from real-time monitoring.

To switch to the charts replaying mode:

- 1 Click **Charts** under the **Monitor** item
- 2 On the **Logged Counters** tab, press the **Add Counters** button on the toolbar to display the **Add Logged Counters** window.
- 3 On the **Data** tab of the **Add Logged Counters** window, press the **Add** button to add any of the available counters in the same way as they are added for real-time monitoring.
- 4 After adding the desired counters, adjust the style of their visualization with the help of the corresponding options on the **Data** tab.

- 5 Go to the **Time** tab of the **Add Logged Counters** window, define the update period, and the time span for which you wish to view the logs for the specified counters:

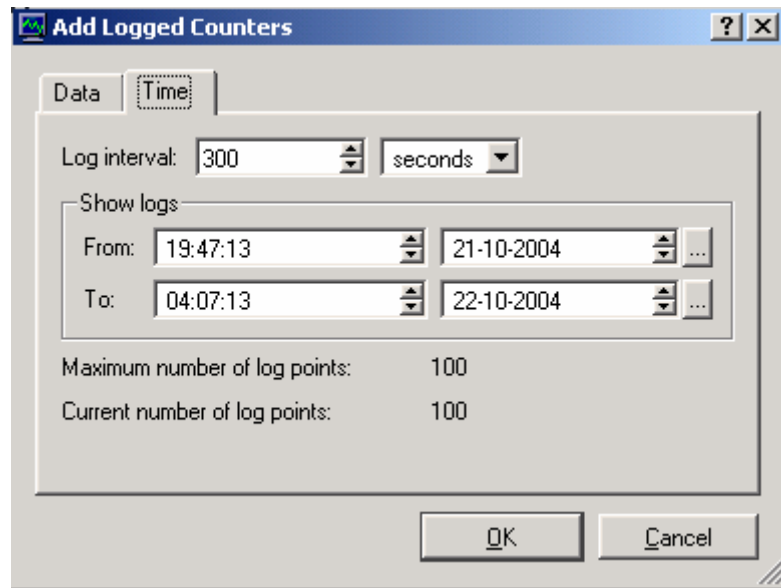


Figure 40: VZMC - Replaying Logs

Please pay attention to the **Maximum number of log points** field, which indicates how many log entries will be shown on the chart. The maximal number is 100. If you have asked for a greater number of entries (this happens in the cases when the update period is too small, or the time span is too large, or both), the update period is automatically lengthened for the time span to contain only 100 entries. This is done to relieve the traffic between the Hardware Node and VZMC.

Using Table Representation

Besides charts, it is possible to monitor many of the Hardware Node or VPS parameters in real time as a list of lines each of which reflects the name and the value of a parameter, as well as the attributes specific for this or that kind of parameters. In such a way, you can view the **Network Rates** and **Processes** groups for a particular Hardware Node, and the **Network Rates**, **Processes**, **Quality of Service (QoS)**, and **Quotas and Usage** groups for a particular Virtual Private Server. Choose any of these groups either in the VZMC main window or in a VPS manager window to see the real-time information about the selected parameters in the form of a table. For example, if you choose **Network Rates** under a HN tree, you may see the following window:

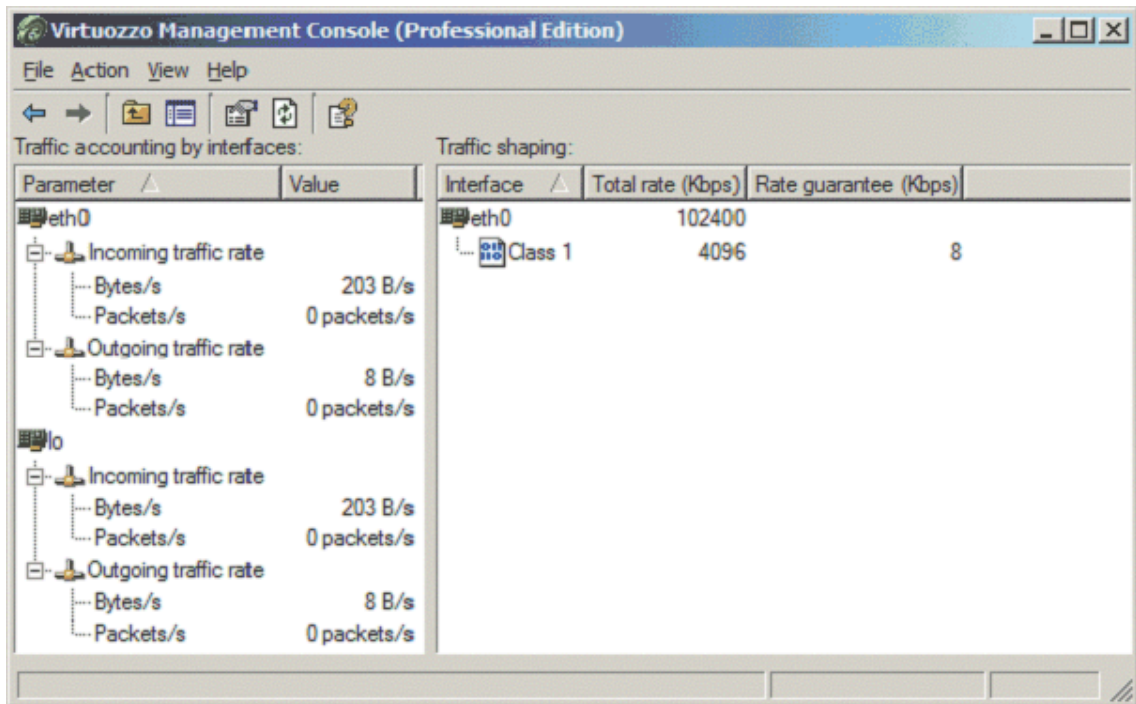


Figure 41: VZMC - Monitoring Traffic Parameters

The **Traffic accounting** table shows the values for the incoming and outgoing traffic rate in bytes per second and packets per second for all the network interfaces present on the Hardware Node. You could view this information as a chart as well.

You can change the time period with which the information is to be refreshed by doing the following:

- 1 Right-click on the **Monitor** item in the Hardware Node or Virtual Private Server main tree and choose **Properties**;
- 2 In the **Monitor Properties** window, select the line corresponding to the group of parameters for which you wish to change the update period;
- 3 Press the **Set Refresh Interval** button on top;
- 4 Enter the new time interval (in seconds) and press **OK**.

Subscribing for Alerts

Alerts constitute an important part of the Virtuozzo monitoring system. They provide a Virtuozzo administrator with a means to keep track of various failures and dangerous situations that might arise in the Virtuozzo system. Bearing in mind that e-mail is not currently the fastest way to deliver the information to the recipient, Virtuozzo alerts can be customized so that you will receive them thru e-mail, thru ICQ, and as SMSs on your mobile phone. You can subscribe to alerts by means of VZMC and of the `vzrmon` utility on the Monitor Node. The kinds of alerts VZMC and `vzrmon` send are different, so you are recommended to use both ways.

Subscribing for VZMC Alerts

VZMC allows you to configure and subscribe for e-mail notifications about resource-overusage system alerts. The subscription for this kind of alerts consists in specifying the e-mail address to send notification to. However, prior to subscribing for alerts, you should provide your e-mail relay server IP address to send e-mail notifications thru. To this effect, you should do the following:

- 1 In VZMC, click on the **Manage Alert Subscription** link on the Hardware Node dashboard to display the **Manage Alert Templates** window;
- 2 On the **Configuration** tab of the **Manage Alert Templates** window, enter an IP address to be used as the mail relay server in the **E-mail relay IP address** field;
- 3 Click OK.

Setting the e-mail relay server IP address is the only required operation that you have to perform before subscribing to alerts. However, there is a number of additional tasks that you may also wish to complete to configure the VZMC alert system to meet your demands:

- Creating your own alert template to specify how your e-mail notifications will look and what information they will contain. By default, VZMC uses a pre-configured notification template. This template includes special placeholders representing special symbols that will be substituted for in the actual message by the actual VPS identifier, parameter name, etc. A list of the main placeholders is given below:
 - `$VEID`: the Virtual Private Server numeric identifier;
 - `$ID`: the name of the QoS parameter (in the actual message, it will be “diskspace”, etc.);
 - `$CURTYPE`: the alert type (at the alert generation moment). The “yellow” alert means that the barrier value of 90% has been reached, the “red” alert – that the value lies in the range from 95% to 100%, and the “black” alert - that the limit value has been hit;
 - `$TIMERANGE`: the alert time (the server time);
 - `$CURVALUE`: the current value of the parameter (at the alert generation moment);
 - `$MAXVALUE`: the maximal value of the parameter during the time when alerts were collected;
 - `$SOFT`: the parameter value barrier;
 - `$HARD`: the parameter value limit.

- Changing the alert policy, i.e. specifying in what way your alerts will be sent. By default, only one alert is sent per subscription and you have to resubscribe to an alert each time after its receiving.

The aforementioned operations can be performed in the **Manage Alert Templates** window used to set your e-mail relay server IP address as well:

- If you are not satisfied with the default alert template, go to the **Templates** tab of the **Manage Alert Templates** window and click on the **Add Alert Template** button to the right of the **Alert template name** field to see the **Alert Template** window. In the displayed window, fill in the provided fields with your own text using any of the above placeholders and press **OK**.
- If you wish to change the default alert policy, you should go to the **Configuration** tab of the **Manage Alert Templates** window and choose one of the following options:
 - **Stop sending.** In this case after having received an alert, you have to resubscribe to it again. This option is selected by default.
 - **Keep sending.** In this case you will get alerts on a permanent basis without having to resubscribe to them each time after their receiving.
 - **Collect alerts before sending for.** In this case alerts will be permanently collected by VZAgent to a special database. This database will be periodically, i.e. with the period specified in the field opposite the option name, checked and if there were any alerts gathered during the set time, the corresponding notification will be sent to your e-mail address. The alert checking time is measured in seconds and can be set either by using the spin button or entering the needed period by hand.

After you have chosen the right option, click **OK** to save the settings.

Now that you have set the e-mail relay server IP address and, if necessary, changed the default alert template and/or default alert policy, you can subscribe for an alert:

- 1 Click on the **Subscribe to Alert** link on the Hardware Node dashboard.
- 2 Select the needed template on the **Alert template name** menu.
- 3 Type the needed e-mail address in the **To** textbox.
- 4 Press the **Subscribe** button.

Subscribing for vzrmond Alerts

vzrmond is a daemon running on your Monitor Node and periodically checking the state of registered Hardware Nodes. The alerts sent by this daemon prove invaluable, for example, when a Hardware Node you administer is down, and you need to instantly receive notifications in these cases. **vzrmond** must be manually configured by editing the `/etc/vzrmond.conf` configuration file. This file contains sections for specifying the host names and/or IP addresses of the Hardware Nodes to be monitored, ICQ UINs, e-mail addresses, and mobile SMS numbers where alerts must be sent, the periodicity of checkups, and a number of parameter values representing the limit upon the reaching of which an alert should be created and sent. All these sections together with their descriptions and recommendations on how to fill them in are given in the **Configuring Virtuozzo** chapter of the **Virtuozzo Reference Guide**. After you have edited the **vzrmond** configuration file, you should restart the daemon to begin receiving alerts.

CHAPTER 8

Managing Services and Processes

This chapter provides information on what services and processes are, the influence they have on the operation and performance of your system, and the tasks they perform in the system.

You will learn how to use the command line utilities and VZMC in order to manage services and processes in Virtuozzo. In particular, you will get to know how you can monitor active processes in your system, change the mode of the `xinetd`-dependent services, identify the VPS ID where a process is running by the process ID, start, stop, or restart services and processes, and edit the service run levels.

In This Chapter

What Are Services and Processes	168
Main Operations on Services and Processes	169
Managing Processes and Services.....	170

What Are Services and Processes

Instances of any programs currently running in the system are referred to as processes. A process can be regarded as the virtual address space and the control information necessary for the execution of a program. A typical example of a process is the `vi` program (on a Linux Node) running on your Hardware Node or inside your VPS(s). Along with common processes, there are a great number of processes that provide an interface for other processes to call. They are called services. In many cases, services act as the brains behind many crucial system processes; they typically spend most of their time waiting for an event to occur or for a period when they are scheduled to perform some task. Many services provide the possibility for other computers on the network to connect to the given one via various network protocols. For example, the `nfs` service provides the NFS server functionality allowing file sharing in TCP/IP networks.

You may also come across the term "daemon" that is widely used in connection with processes and services. This term refers to a software program used for performing a specific function on the computer system and is usually used as a synonym for "service". It can be easily identified by "d" at the end of its name. For example, `httpd` (short for HTTP daemon) represents a software program that runs in the background of your system and waits for incoming requests to a web server. The daemon answers the requests automatically and serves the hypertext and multimedia documents over the Internet using HTTP.

When working with services, you should keep in mind the following. During the lifetime of a service, it uses many system resources. It uses the CPUs in the system to run its instructions and the system's physical memory to hold itself and its data. It opens and uses files within the filesystems and may directly or indirectly use certain physical devices in the system. Therefore, in order not to damage your system performance you should run only those services on the Hardware Node that are really needed at the moment.

Besides, you should always remember that running services in the Host OS is much more dangerous than running them in Virtual Private Servers. In case violators get access to one of the VPSs thru any running service, they will be able to damage only the VPS where this service is running, but not the other VPSs on your Hardware Node. The Hardware Node itself will also remain unhurt. And if the service were running on the Hardware Node it would damage both the Hardware Node and all the VPSs residing on it. Thus, you should make sure that you run only those services on the Hardware Node that are really necessary for its proper functioning. Please launch all additional services you need at the moment inside separate VPSs. It will significantly improve your system safety.

Note: In VZMC, you can view all available services by clicking on the **Services** folder item in the tree below the Hardware Node name or the Virtual Private Server name or clicking on the **Manage Unix Services** link on the corresponding summary page.

When working with the command line, you can use the `vzps` or `vztop` utilities to display all the processes that are currently running in your system.

Main Operations on Services and Processes

The ability to monitor and control processes and services in your Virtuozzo system is essential because of the profound influence they have on the operation and performance of your whole system. The more you know about what each process or service is up to, the easier it will be to pinpoint and solve problems when they creep in.

The most common tasks associated with managing services in the Host Operating System of the Hardware Node or inside a Virtual Private Server are starting, stopping, enabling, and disabling a service. For example, you might need to start a service in order to use certain server-based applications, or you might need to stop or pause a service in order to perform testing or to troubleshoot a problem.

For `xinetd`-dependent services, you do not start and stop but enable and disable services. The services enabled in this way are started and stopped on the basis of the corresponding state of the `xinetd` daemon. Disabled services are not started whatever the `xinetd` state.

The services management is mostly disabled for the Hardware Node. Practically all the services are read-only, you are able to view the information but you cannot perform any operation on them. The reason is that many Red Hat packages determine a successful stop by looking up all the processes with a specified name. If such processes exist elsewhere, they are killed with the terminate signal. Thus, all the like services in all the HN Virtual Private Servers might be accidentally shut down because of this.

However, there are some services that can be managed by a number of administrative tools offered in Virtuozzo. These tools allow a service to be managed and configured either by means of special Linux command-line utilities or via VZMC. You can do it either locally or from any computer connected on the network. Besides, you can manage all the processes and services thru VZPP (Virtuozzo Power Panels). All the necessary information on managing services and operations in VZPP is provided in the comprehensive online help system and the user's manual VZPP is supplied with.

As for processes, such utilities as `vzps`, `vztop`, `vzpid` enable you to see what a process is doing and to control it. Sometimes, your system may experience problems such as slowness or instability, and using these utilities should help you improve your ability to track down the causes. It goes without saying that in Virtuozzo you can perform all those operations on processes you can do in the common Linux system, for example, kill a process by sending a terminate signal to it.

In Virtuozzo, you can manage services and processes using both the command line and VZMC. Further in this chapter, both methods are described.

Managing Processes and Services

In Virtuozzo, services and processes can be managed by using both the command line and VZMC. In the command line, you can manage the corresponding processes and services by using the following utilities:

- `vzps`,
- `vzpid`,
- `vztop`, and
- `vzsetxinetd`.

With their help you can perform the following tasks:

- Print information about active processes on your Hardware Node;
- Display the processes activity in real time;
- Change the mode of the services that can be either `xinetd`-dependent or standalone;
- Identify the Virtual Private Server ID where a process is running by the process ID.

Virtuozzo Management Console allows you to manage the services present in the Host Operating System of the Hardware Node or in a Virtual Private Server. It allows you to monitor (and partially configure) the services of the Host Operating System at the Hardware Node(s). By using VZMC, you can start, stop, restart a service, or edit its run levels. Below in this chapter detailed information on all those tasks that can be performed by means of the command line and VZMC is given.

Viewing Active Processes and Services

The `vzps` utility can be run on the Hardware Node just as the standard Linux `ps` utility. It provides certain additional functionality related to monitoring separate Virtual Private Servers running on the Node, namely, you can pass the `-E` switch to the `vzps` utility to display the Virtual Private Server IDs where the processes are running.

`vzps` prints information about active processes on your Hardware Node. If you issue the `vzps` command alone, it will list only those processes that are running on the current terminal. Below is an example output of the `vzps` run:

```
$ ps
  PID TTY          TIME CMD
 4684 pts/1    00:00:00 bash
27107 pts/1    00:00:00 vzps
```

Currently, the only processes assigned to the user/terminal are the `bash` shell and the `vzps` command itself. In the output, the `PID` (Process ID), `TTY` , `TIME` , and `CMD` fields are contained. `TTY` denotes which terminal the process is running on, `TIME` shows how much CPU time the process has used, and `CMD` is the name of the command that started the process.

Note: Starting from Virtuozzo 3.0, the IDs of the processes running inside Virtual Private Servers and displayed by running the `vzps` command on the Hardware Node does not coincide with the IDs of the same processes shown by running the `ps` command inside these Virtual Private Servers.

As you can see, the standard `vzps` command just lists the basics. To get more details about the processes running on your Hardware Node, you will need to pass some command line arguments.

Passing `vzps` the commonly used `aux` arguments displays processes started by other users (`a`), processes with no terminal or one different from yours (`x`), the user who started the process and when it began (`u`). Besides, you can pass `vzps` the `-E` switch, which is specific for Virtuozzo, to view the VPS IDs where the processes are running.

```
[root@ts23 root]# vzps aux -E
VEID USER  PID %CPU %MEM  VSZ  RSS TTY     STAT START  TIME COMMAND
 0   root    1  0.0  0.0 1516  128 ?       S    Jul14  0:37 init
 0   root    5  0.0  0.0    0    0 ?       S    Jul14  0:03 [ubstatd]
 0   root    6  0.0  0.0    0    0 ?       S    Jul14  3:20 [kswapd]
123  #27     7  0.0  0.0    0    0 ?       S    Jul14  0:00 [bdf flush]
 0   root    9  0.0  0.0    0    0 ?       S    Jul14  0:00 [kinoded]
 0   root  1574  0.0  0.1  218  140 pts/4   S    09:30  0:00 -bash
```

There is a lot more information now. The fields `VEID` , `USER` , `%CPU` , `%MEM` , `VSZ` , `RSS` , `STAT` , and `START` have been added. Let's take a quick look at what they tell us.

First, you see the `VEID` field which denotes the Virtual Private Server the process belongs to. Please note that the VPS with `ID = 0` is the Hardware Node itself, and the VPS with `ID = 1` marks your Service VPS. Besides, all processes are displayed now, not just the ones running on your terminal. The `USER` field shows you which user initiated the command. Many processes begin at system start time and often list `root` or some system account as the `USER` . Other processes are, of course, run by individuals.

The %CPU, %MEM, VSZ, and RSS fields all deal with system resources. First, you can see what percentage of the CPU the process is currently utilizing. Along with CPU utilization, you can see the current memory utilization and its VSZ (virtual memory size) and RSS (resident set size). VSZ is the amount of memory the program would take up if it were all in memory; RSS is the actual amount currently in memory. Knowing how much a process is currently eating will help determine if it is acting normally or has spun out of control.

You will notice a question mark in most of the TTY fields in the `vzps aux` output. This is because most of these programs were started at boot time and/or by initialization scripts. The controlling terminal does not exist for these processes; thus, the question mark. On the other hand, the `bash` command has a TTY value of `pts/4`. This is a command being run from a remote connection and has a terminal associated with it. This information is helpful for you when you have more than one connection open to the machine and want to determine which window a command is running in.

STAT shows the current status of a process. In our example, many are sleeping, indicated by an `S` in the STAT field. This simply means that they are waiting for something. It could be user input or the availability of system resources. The other most common status is `R`, meaning that it is currently running.

For detailed information on all `vzps` parameters, output fields, states of processes, etc., consult the Linux Administrator's Guide or the corresponding manual pages.

In its turn, Virtuozzo Management Console allows you to monitor the services present in the Host Operating System of the Hardware Node or in a Virtual Private Server. Click on the Services folder item in the tree below the Hardware Node name. A list of the Host OS or VPS OS services should appear in the right pane:

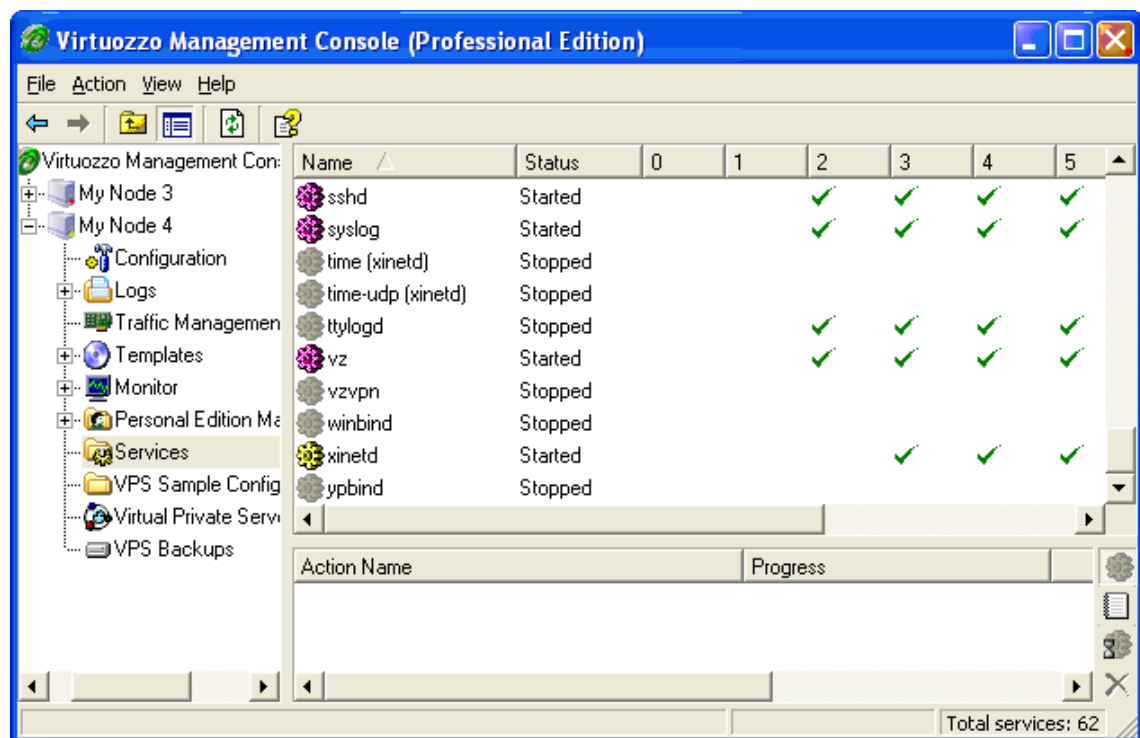


Figure 42: VZMC - Viewing Services

The way the services are colored reflects the importance of a service for Virtuozzo: pink icons are for services that are critical for Virtuozzo and yellow icons are for services that are not that critical.

Running services are indicated with bright icons. Stopped services have shaded icons. The **Status** column of the table duplicates this information in the text form. The default run levels of services are ticked off in the corresponding table columns.

To facilitate working with services, you can sort them by different parameters: their name, status, etc. Just click the column with the appropriate name to put services in the desired order.

Monitoring Processes in Real Time

The `vztop` utility is rather similar to `vzps` but is usually started full-screen and updates continuously with process information. This can help with programs that may infrequently cause problems and can be hard to see with `vzps`. Overall system information is also presented, which makes a nice place to start looking for problems.

The `vztop` utility can be run on the Hardware Node just as the standard Linux `top` utility. The only features that distinguish the `vztop` utility from `top` are the following:

- `vztop` allows you to use the `E VEID` option that monitors only the processes belonging to the VPS whose processes you wish to display;
- you can use the `e` interactive command to temporarily view/hide the VPSIDs where the processes are running;
- you can use the `E` interactive command to set the filter on the VPSID field that helps you display only the processes belonging to the given VPS.

The `vztop` utility usually has an output like the following:

```
[root@ts23 root]# vztop E 1
17:54:03 up 20 days, 23:37, 4 users, load average: 2.13, 1.89, 1.75
305 processes: 299 sleeping, 3 running, 3 zombie, 0 stopped
CPU0 states: 20.1% user 51.2% system 0.0% nice 0.0% iowait 28.1% idle
CPU1 states: 21.2% user 50.0% system 0.0% nice 0.0% iowait 28.1% idle
Mem: 1031088k av, 969340k used, 61748k free, 0k shrd, 256516k buff
      509264k active,          330948k inactive
Swap: 4056360k av, 17156k used, 4039204k free 192292k cached
  PID USER PRI NI SIZE RSS SHARE VEID STAT %CPU %MEM TIME CPU COMMAND
1955 root 14 0 1292 404 376 1 S 0.9 0.0 0:10 0 vncp
1131 root 9 0 104 80 56 1 S 0.0 0.0 0:05 1 init
1774 root 9 0 744 608 576 1 S 0.0 0.0 0:00 0 sshd
1784 root 9 0 480 384 384 1 S 0.0 0.0 0:00 0 xinetd
1832 mail 10 0 536 456 396 1 S 0.0 0.0 0:00 0 exim
1964 #100 9 0 560 440 384 1 S 0.0 0.0 8:29 1 thttpd
```

As you can see, `vztop` provides an ongoing look at the processor activity in real time (the display is updated every 5 seconds by default, but you can change that with the `d` command-line option or the `s` interactive command). It displays a list of the most CPU-intensive tasks on the system and can provide an interactive interface for manipulating processes. It can sort the tasks by CPU usage, memory usage, and runtime. You can see that at the moment only those processes are displayed which are running on the VPS whose ID is equal to 1. Besides, most features can be selected by an interactive command, for example, the `e` and `E` commands described above.

For more information on all `vztop` parameters, consult the Linux Administrator's Guide or the corresponding man pages. Besides, you can find information on some fields in the Viewing Active Processes subsection (on page 171).

In VZMC, you can view those processes that are currently running on your Hardware Node and/or inside your Virtual Private Server(s). To display the processes, click the Hardware Node name where you wish to monitor processes and then select **Monitor -> Processes**. A list of the Host OS or VPS OS processes should appear in the right pane:

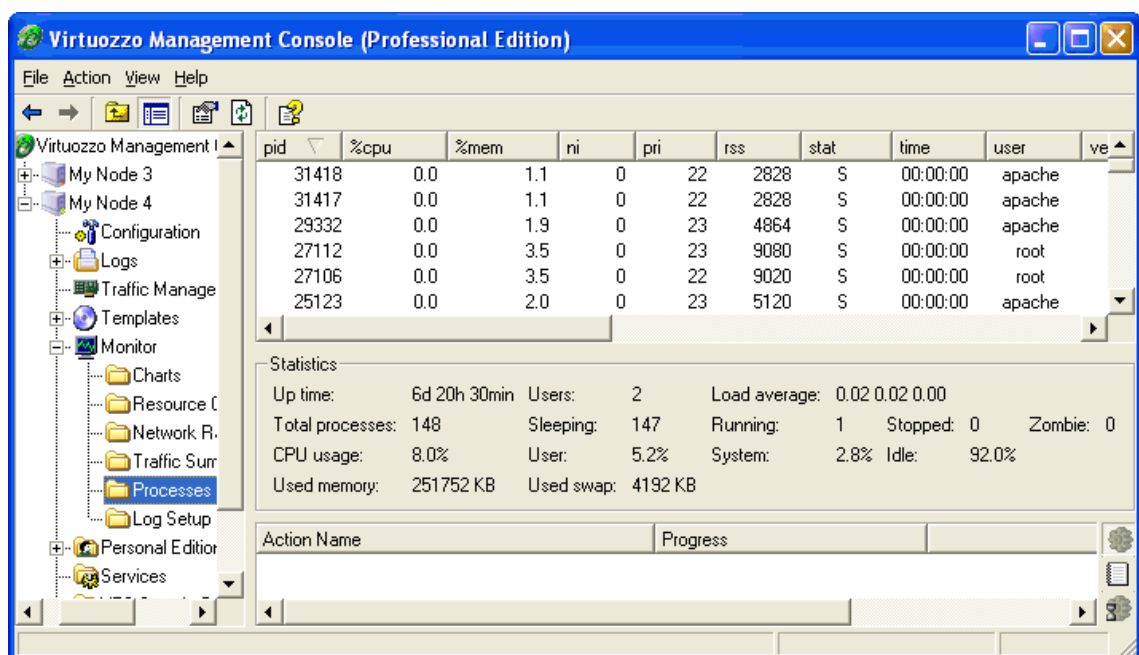


Figure 43: VZMC - Monitoring Active Processes

The column names and their description is presented in the table below:

Column name	Description
pid	The identifier of the process.
user	The user who has launched the process.
pri	The kernel scheduling priority for the process.
ni	The 'nice' parameter; weights the overall scheduling priority for the process.
rss	Number of resident pages for the swap-out guarantee (the resident set size).
stat	The process current status. Can be 'R' (running), 'S' (sleeping, waiting for 'wake-up call'), 'D' (uninterruptable sleep), 'Z' (zombie, waiting for parent process), 'T' (stopped or traced). Sometimes the second symbol may appear: 'W' (process swapping), 'N' ('niced' process), 'L' (process has pages locked into memory). If the < sign is displayed after the status, it means that this information was returned by vzagent, and vzagent got this information from the ps tool.
%cpu	The CPU time in percents used by the process.
%mem	The memory used by the process.
time	The total CPU time the process has used.
command	The command that invoked the process.
vpsid	The ID of the VPS where the process is running.

To view the processes inside a Virtual Private Server, double-click on its name and select **Monitor -> Processes**.

Note: Starting from Virtuozzo 3.0, the IDs of the processes running inside your Virtual Private Servers displayed by selecting **Monitor -> Processes** on the Hardware Node does not coincide with the IDs of the same processes shown when opening the VPS Manager window and selecting **Monitor --> Processes**.

You can send different signals to process by right-clicking a process and selecting the corresponding signal on the pop-up menu.

Changing Services Mode

`xinetd` is a service used to start and stop a variety of data communication services. `xinetd` starts on the HN startup and waits for a connection request from a remote client that wants to connect to the server. There can be a number of remote clients in the network, and each of them can use different network protocols to establish connection to the server. In order not to run all network services responsible for this or that protocol, which will negatively influence the system performance, the system starts only the `xinetd` service. This service controls all other network services and, at the connection time, it starts the corresponding service to process this connection. In such a way, `xinetd` saves system resources allowing you to run only those network services in the system that are really needed at the moment.

The `vzsetxinetd` utility allows you to switch Virtual Private Server services between the standalone and `xinetd` mode. The services that can be either standalone or dependent on `xinetd` in the current release of Virtuozzo are `sendmail`, `sshd`, `proftpd`, and `courier-imap`. Whereas they are `xinetd`-dependent by default in order to consume less resources, you may want to make them standalone due to the following reasons:

- The CPanel application does not recognize `sshd` if it is dependent on `xinetd`;
- `sendmail` does not process some rules correctly if it is dependent on `xinetd`;
- A number of control panel applications and some others are not able to manage `xinetd`-based services at all.

The `courier-imapd`, `courier-imapds`, `courier-pop3d`, and `courier-pop3ds` services are provided by the `courier-imap` service, thus `vzsetxinetd` can manage these services via the `courier-imap` service.

Let us assume that you wish to check the mode of the `sendmail` service and set it to standalone if it is in the `xinetd` mode. First, you should check the current status of the `sendmail` service. To this effect, type the following command in the command line:

```
[root@ts23 root]# vzsetxinetd -s 222 sendmail
```

where 222 is the Virtual Private Server ID, `sendmail` denotes the name of the corresponding service, and the `-s` option gets the status of the `sendmail` service of the VPS with ID 222. The output will tell you if this service has the standalone or `xinetd` mode:

```
sendmail is xinetd service
```

In our case it is in the `xinetd` mode. Now you can change the mode of the `sendmail` service to standalone. To make it standalone, type the following line:

```
[root@ts23 root]# vzsetxinetd 222 sendmail off
sendmail is standalone service
```

where `off` specifies that the `sendmail` service should be set to the standalone mode. The output confirms that the `sendmail` service is now standalone.

For more information on the `vzsetxinetd` utility, please consult the corresponding man pages or turn to the [Virtuozzo Reference Guide](#).

Note: You cannot use the `vzsetxinetd` utility to change the mode of the `xinetd`-dependent services in Virtual Private Servers where the Debian 3.0 OS template is installed.

Determining VPS Identifier by Process ID

Each process is identified by a unique PID (process identifier), which is the entry of that process in the kernel's process table. For example, when you start Apache, it is assigned a process ID. This PID is then used to monitor and control this program. The PID is always a positive integer. In Virtuozzo you can use the `vzpid` (retrieve process ID) utility to print the Virtual Private Server ID the process with the given id belongs to. Multiple process IDs can be specified as arguments. In this case the utility will print the Virtual Private Server number for each of the processes.

The typical output of the `vzpid` utility is shown below:

```
[root@ts23 root]# vzpid 12
Pid      VEID      Name
12        4         init
```

In our example the process with the identifier 12 has the name 'init' and is running in the Virtual Private Server with ID = 4.

Note: You can also display the VPS ID where the corresponding process is running by using the `vzps` utility.

Starting, Stopping, and Restarting Services

Virtuozzo Management Console allows you to manage the services present in the Host Operating System of the Hardware Node or in a Virtual Private Server. Click on the **Services** folder item in the tree below the Hardware Node name or the Virtual Private Server name or click on the **Manage Unix Services** link on the corresponding summary page. A list of the Host OS or VPS OS services should appear in the right pane:

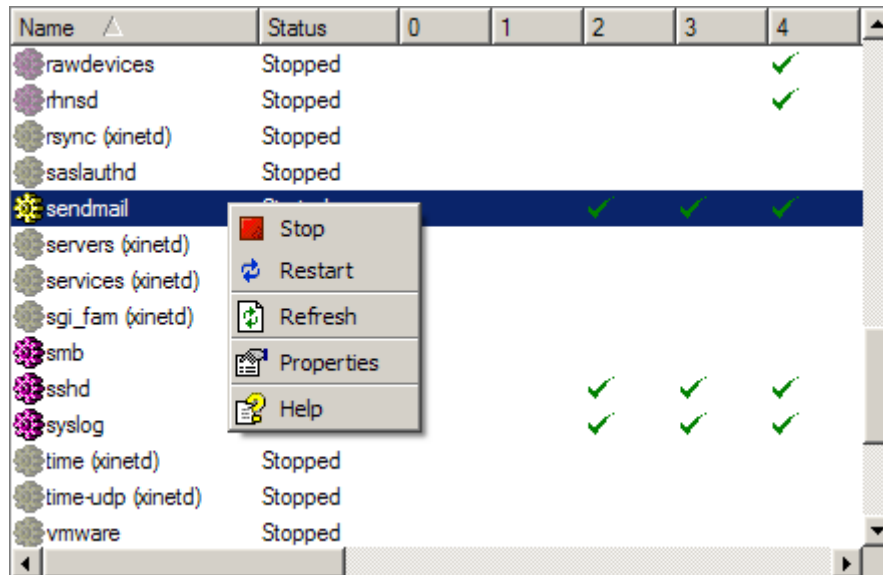


Figure 44: VZMC - Managing Processes and Services

To start, stop, or restart a service, select its line in the table and either use the pop-up menu or the buttons on the toolbar. For `xinetd`-dependent services (the services having `xinetd` in parentheses beside their name), you do not start and stop but enable and disable services. The services enabled in this way are started and stopped on the basis of the corresponding state of the `xinetd` daemon. Disabled services are not started whatever the `xinetd` state.

To edit the default run levels for the service, use the **Properties** item from the pop-up menu or just double-click on the service name within the list. When the **Properties** dialog is open, mark the checkboxes of the run levels on which the service will start automatically. Click the **OK** button to apply your settings. If the service is dependent on `xinetd`, you cannot choose its run levels, as the latter are determined by the `xinetd` daemon. Besides, you cannot change run levels for certain services, which means that they are critical and you are not allowed to change their run levels.

You can also manage (i.e. start, stop, and restart) services by using the command line. For example, you wish to start the `httpd` service. To this effect, you should type the following command:

```
[root@vps222 /]# service httpd start
```

where `service` is the standard Linux command, `httpd` denotes the name of the corresponding service, and `start` is the command that will launch this service. In order to check that the `httpd` service was successfully launched, you can either type the following Linux command:

```
[root@vps222 /]# service httpd status
```

or use the `vzps` utility when working on your Hardware Node or the `ps` utility when working inside your VPS(s) and passing them the `x` argument. The output will tell you if the `httpd` service is running in your system or not.

CHAPTER 9

Name-Based Hosting for Virtuozzo

Virtuozzo name-based hosting is a VPS-related implementation of a well-known technique of name-based web hosting, implemented, for example, in the Apache web server. Name-based web hosting is widely used alongside of IP-based web hosting. Unlike the latter, it refers to the process of running multiple "virtual" web servers on a single IP address of the host computer. Using this technique, a single computer can host thousands of independent web sites.

Virtuozzo extends the notion of name-based web hosting to name-based hosting as such, i.e. running a slew of full-fledged Virtual Private Servers sharing a single external IP address on a single Hardware Node, these Virtual Private Servers having all the characteristics of independent physical servers (not only of web servers) accessible from without by their hostnames. This means running a full set of network services inside any such Virtual Private Server.

In This Chapter

Name-Based Hosting Overview	181
Basic Configuration	183
Advanced Configuration	184
Setting Up SMTP Disk Quota	187
Learning Current Name-Based Hosting Status	188

Name-Based Hosting Overview

Virtuozzo name-based hosting enables the Hardware Node administrator to provide a single external IP address for all or a number of VPSs hosted on that Node. All network requests received by the Node are redirected to the corresponding VPS according to the domain name supplied by the requesting client, enabling the Hardware Node to differentiate among multiple Virtual Private Servers on the one IP address. This process is visualized in the following scheme:

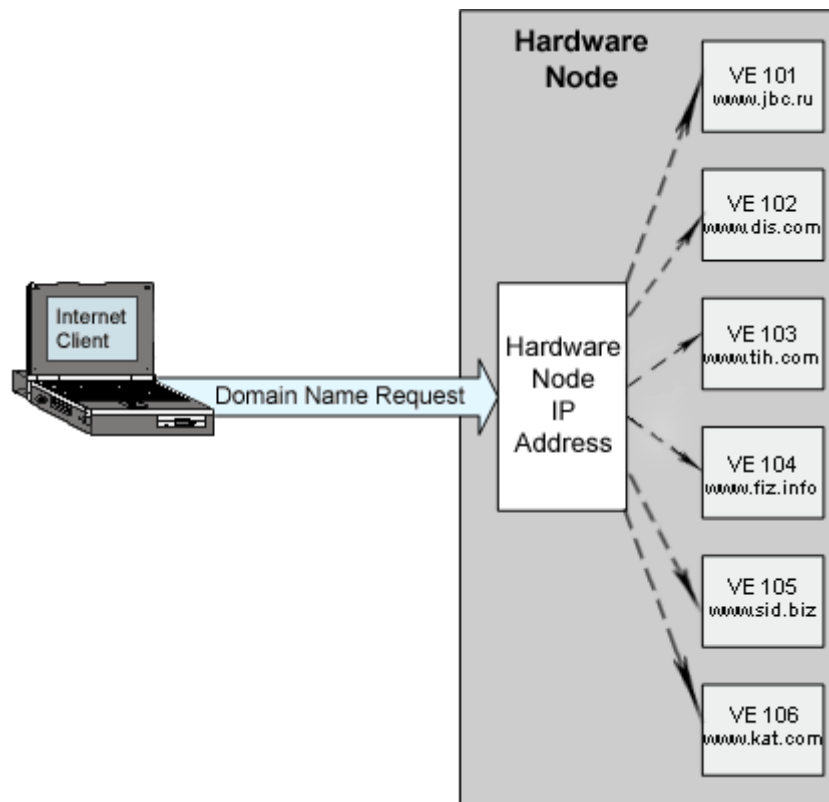


Figure 45: Name-Based Hosting

A client tries to reach the VPS server by its domain name. This domain name is registered with the DNS servers and points to one of the Hardware Node external IP addresses assigned for name-based hosting. Indicating only the Hardware Node IP address in the client's request is not enough, as this address is common for a number of name-based VPSs. As to the IP address of the VPS proper, it is internal and, therefore, is not used to receive requests from the outside of the local network.

Note: The Hardware Node external IP address assigned for name-based hosting should be directly associated with one of the physical network interfaces of the Node. It goes without saying that it should be reachable from the outside world, i.e., the administrator should configure the network so that the packets destined to that address reach the Node somehow. There are no other restrictions as to the configuration of this address.

When the client's request reaches the Node IP address, it is redirected to the appropriate VPS thanks to the VPS domain name supplied by the client. In case of the HTTP protocol, this domain name is specified in the `Host: domain_name` header in the request. When the connection is established by means of the other protocols, the domain name is resolved by the provided credentials. In this way one and the same external IP address of the Hardware Node can be used to provide network services for any number of Virtual Private Servers.

Important! Though name-based hosting allows a multitude of VPSs to be accessible from the outside by one external IP address, it has nothing to do with the ability of these VPSs to access hosts of the outer world. To enable the VPSs, which have only internal IP addresses, to access the Internet, SNAT (Source Network Address Translation) or IP masquerading should be configured on the Hardware Node. This is ensured by the standard Linux `iptables` utility. To perform a simple SNAT setup, you should do the following:

- 1 Execute the following command on the Hardware Node:

```
# iptables -t nat -A POSTROUTING -s scr_net -o eth0 -j SNAT --to ip_address
```

where `scr_net` is a range of IP addresses of name-based VPSs to be translated by SNAT, and `ip_address` is the external IP address of your Hardware Node. Multiple rules are allowed, for example, in case you wish to specify several ranges of IP addresses. If you are using a number of physical network interfaces on the Node, you may need to specify a different interface for outgoing connections, e.g. `-o eth2`.

To make all IP addresses to be translated by SNAT (not only the ones of name-based VPSs), you should issue the following command:

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to ip_address
```

- 2 Add the following string to the `/etc/modules.conf` file: `"options ip_conntrack ip_conntrack_enable_ve0=1"`. For example, you can do it by opening the `/etc/modules.conf` file by means of the `vi` text editor and entering the needed string.
- 3 Make the `ipt_conntrack` and `iptable_nat` modules to be loaded on the Hardware Node startup. To this effect, provide the names of these modules as the value of the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file on the Node.

After you have added the `ipt_conntrack` and `iptable_nat` modules, you should either reboot your Hardware Node or stop Virtuozzo, restart the `iptables` utility, and start Virtuozzo anew for the changes to take effect. In the latter case you should do the following:

```
# service vz stop
# /etc/rc.d/init.d/iptables restart
# service vz start
```

Note: The `iptables` utility allows to set up more complex rules for Network Address Translation, involving various protocols and ports. If you wish to get more information on this, consult the numerous Internet sites (e.g. <http://www.netfilter.org>) and tutorials (e.g. <http://www.yolinux.com/TUTORIALS/LinuxTutorialIptablesNetworkGateway.html>) that are devoted to this issue.

Configuring Virtuozzo for providing name-based hosting services is covered in the following sections.

Basic Configuration

To enable name-based hosting for any VPS on the Hardware Node, you should:

- 1 Make sure the hostname(s) you are planning to assign to the VPS are properly registered with the DNS servers and point to that IP address of the Hardware Node which you are going to use for name-based hosting.
- 2 Modify the global Virtuozzo configuration file (`/etc/sysconfig/vz`) to
 - Enable name-based hosting for the given Hardware Node:

```
NBH=yes
```

- 3 Modify the name-based hosting configuration files (`ftp.conf`, `http.conf`, `pop3.conf`, `smtp.conf`) located in the `/etc/vznbh.d` directory and related to the major network protocols to:
 - Enable the name-based hosting service for the given protocol;
 - Indicate which IP address and TCP port of the Node is to serve for the name-based connections by the given protocol. For example:

```
EXTERNAL_IP="200.200.200.200"
PROTOCOL="HTTP"
PORT="80"
DISABLED="no"
```

Note: You do not have to specify the IP address and TCP port for the `smtp` protocol.

- 4 Add the following string to the `/etc/modules.conf` file: `"options ip_conntrack ip_conntrack_enable_ve0=1"`. For example, you can do it by opening the `/etc/modules.conf` file by means of the `vi` text editor and entering the needed string.
- 5 Make the `ipt_conntrack` and `iptables_nat` modules to be loaded on the Hardware Node startup. To this effect, provide the names of these modules as the value of the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file on the Node.

After you have added the `ipt_conntrack` and `iptables_nat` modules, you should either reboot your Hardware Node or stop Virtuozzo, restart iptables, and start Virtuozzo anew for the changes to take effect. In the latter case you should do the following:

```
# service vz stop
# /etc/rc.d/init.d/iptables restart
# service vz start
```

- 6 Modify the corresponding VPS configuration file (`/etc/sysconfig/vz-scripts/<vpsid>.conf`) to
 - Enable the name-based hosting for the given VPS (the `NBH_ENABLE` parameter);
 - Indicate the internal IP address of the VPS to be used for name-based hosting (the `NBH_IP` parameter); as a rule, this address coincides with the value of the `IP_ADDRESS` parameter;
 - Define the hostname(s) for the VPS (the `NBH_HOSTNAME` parameter);
 - Enable/disable certain services for name-based hosting (the `NBH_SERVICE` parameter). For example:

```
NBH_ENABLE=yes
NBH_IP="192.168.0.100"
NBH_HOSTNAME="www.myhost.com www.myhost2.info"
NBH_SERVICE="smtp http pop3"
```

The values of the NBH_SERVICE parameter should coincide with the names of the files in the /etc/vznbh.d directory without the .conf suffix.

You might as well use the vzctl utility to set all these parameters:

```
# vzctl set 101 --nbh_enable yes --nbh_ip 192.168.0.100 --
nbh_hostname_add www.myhost.com --nbh_hostname_add www.myhost2.info --
nbh_service_add smtp --nbh_service_add http --nbh_service_add pop3 --
save
```

After you have thus modified all the needed configuration files, you should restart the Virtuozzo name-based hosting service for the changes to take effect:

```
# service vz nbhrestart
```

If you have configured everything correctly, VPS 101 is now likely to be accessible from the outside by the http, smtp, and pop3 protocols by the specified hostnames.

Note: When connecting to a name-based VPS with an FTP or POP3 account, the client should provide the user name in the user@host format, i.e. append the hostname to the user name proper. For example: root@myhost.com.

Inasmuch as not all FTP clients allow to use the @ sign as part of the user name, it is possible to replace this sign with the % symbol when connecting to name-based VPSs by the FTP protocol. For example: root%myhost.com.

Advanced Configuration

Virtuozzo name-based hosting can be further flexibly configured to meet certain specific needs of the Hardware Node administrator. For example, you may wish to:

- Configure certain services on non-standard additional ports;
- Assign different ports for one and the same service for different VPSs;
- Set up more complicated rules by configuring a few of the services on an additional IP address;
- Configure name-based VPSs to use different external IP addresses;
- etc.

If you have in mind any of these or similar tasks, you should perform the following steps:

- 1 Define a full set of IP addresses and ports for various services that you are planning to use for name-based hosting. This is done by creating new .conf files in the /etc/vznbh.d directory. The basic http.conf and other files discussed in the previous section are an example of how such files shall be created. You may create new files for declaring additional IP addresses to be used for name-based hosting and additional TCP ports for various services on any IP address declared in this way. These files should have the .conf suffix and should not begin with a dot.

- 2 For each particular VPS, specify the IP address/TCP port/protocol sets the given VPS is allowed to use by specifying the corresponding `.conf` files created at Step 1. This is done in the corresponding VPS configuration file (`/etc/sysconfig/vz-scripts/<vpsid>.conf`).

These steps together with an example of advanced configuration are described in the following subsections.

Defining Global Protocol Parameters

The default Hardware Node IP addresses and TCP ports to be used for name-based connections by all the major protocols are set up by the default configuration files located in the `/etc/vznbh.d` directory. To define an additional IP address or an additional port for this or that protocol to be used in name-based hosting, you should create an additional file in this directory. There are two requirements for this file:

- 1 The name of this file should not begin with a dot and should have the `.conf` suffix;
- 2 The structure of this file should correspond to the structure of the default files.

In this way you may define as many extra files as you deem reasonable. For example:

```
# cat /etc/vznbh.d/http.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.200"
PORT="80"
DISABLED="no"

# cat /etc/vznbh.d/http_sec.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.201"
PORT="80"
DISABLED="no"

# cat /etc/vznbh.d/http_sec8080.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.201"
PORT="8080"
DISABLED="no"
```

Additional IP addresses and TCP ports can thus be defined for the `http`, `ftp`, and `pop3` protocols. As to the `smtp` protocol, it does not need any.

Setting Up VPS Permissions

After you have thus defined all the Hardware Node IP addresses to be used for name-based hosting and all the TCP ports to be used for various services on particular IP addresses, you may allow or forbid any VPS to use any such IP address/ TCP port. This is done in the corresponding VPS configuration file (`/etc/sysconfig/vz-scripts/<vpsid>.conf`). You should specify which services on which IP addresses and TCP ports of those defined in the global protocol configuration files the given VPS is allowed to use. The parameter format is as follows:

```
NBH_SERVICE="file file ..."
```

For example:

```
NBH_SERVICE="http http_sec http_sec8080 smtp"
```

Mind that the name of the file should coincide exactly with the corresponding file from the `/etc/vznbh.d` directory taken without the `.conf` suffix.

Allowing various IP addresses and TCP ports combinations to be used for various name-based VPSs, you are able to create an unlimited number of name-based hosting configurations: all depends on your specific needs.

Example

Task: Let us assume that you wish to define two external IP addresses to be used for name-based hosting on your Hardware Node. Then, you want the `http` service to be accessible on port 80 and port 8080 on the first IP address, and on port 80 on the second one. One group of VPSs hosted on your Node are allowed to use both 80 and 8080 ports on the first IP address only, another group of VPSs - ports 80 on both IP addresses, and still another group of VPSs - only port 80 on the second IP address.

Solution:

- 1 Create three files defining all the corresponding IP addresses and TCP ports in the `/etc/vznbh.d` directory:

```
# cat /etc/vznbh.d/http.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.200"
PORT="80"
DISABLED="no"

# cat /etc/vznbh.d/http_8080.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.200"
PORT="8080"
DISABLED="no"

# cat /etc/vznbh.d/http_201_80.conf
PROTOCOL="HTTP"
EXTERNAL_IP="200.200.200.201"
PORT="80"
DISABLED="no"
```

- 2 For the first group of VPSs, define the following parameter in their corresponding configuration files:

```
NBH_SERVICE="http http_8080"
```

- 3** For the second group of VPSs, define the following parameter in their corresponding configuration files:

```
NBH_SERVICE="http http_201_80"
```

- 4** For the third group of VPSs, define the following parameter in their corresponding configuration files:

```
NBH_SERVICE="http_201_80"
```

The task is solved. Note that as a rule you allow VPSs to use other services as well so, in reality, the `NBH_SERVICE` parameter includes more file names. You may configure the `ftp` and `pop3` services in completely the same manner.

Setting Up SMTP Disk Quota

To deliver mail to Virtual Private Servers with name-based hosting enabled, Virtuozzo uses the `exim` package installed inside the Service VPS. If any mail destined to a VPS cannot be delivered therein (e.g. the VPS is stopped), it is stored in the Service VPS until the destination VPS is ready to accept it.

To avoid cluttering disk space with unclaimed mail, there are restrictions on the size of mail messages any VPS is entitled to temporarily keep in the Service VPS. You can learn this quota from the `/etc/exim/quota.limits` file inside the Service VPS. The following line in this file defines the SMTP disk quota:

```
*:5242880
```

`*` designates that the quota applies to any Virtual Private Server, and 5242880 is the number of bytes reserved for the mail of any particular VPS (around 5 Mb). You can increase or decrease this number at your discretion, as well as set up different quota limits for any particular Virtual Private Server. For example, to have the default VPS mail quota equal 10 Mb, and reduce this quota to 5 Mb for VPS 101, enter the following lines into the above file and save it. It will then look like the following:

```
*:10485760
101:5242880
```

Naturally, you should make sure that you have enough disk space in the Service VPS. The necessary disk space may be calculated by multiplying the SMTP disk quota by the number of name-based VPSs you are planning to host on the Node.

Learning Current Name-Based Hosting Status

Virtuozzo provides the standard Linux `/proc` file system interface for getting to know the current status of various parameters related to name-based hosting. You may consult the following files located in the `/proc/proxy` directory:

- `clients` - to learn about the currently connected clients;
- `hostnames` - to learn what name-based VPSs have what hostnames;
- `protocols` - to learn about the currently enabled protocols for name-based hosting;
- `services` - to learn about all the name-based services defined by separate configuration files in the `/etc/vznbh.d` directory;
- `targets` - to learn the current name-based hosting statistics on the target VPSs.

CHAPTER 10

Keeping Your Virtuozzo System Up-to-Date

Being a virtualization solution, the Virtuozzo software modifies the host operating system on a rather low level, even including kernel modification. With this in mind, the Virtuozzo Hardware Node administrator is supposed to understand what are the ways to keep the Virtuozzo system up-to-date by applying all the latest security fixes and other updates.

The components to be kept up-to-date are the following:

- the host OS software;
- the Virtuozzo software, and
- the VPSs created on the Node.

In This Chapter

Updating Host OS Software.....	189
Updating Virtuozzo Software	191
Updating Software Inside Virtual Private Servers	191

Updating Host OS Software

Beginning with Virtuozzo 3.0, it is safe to use the native Linux distributions updaters (`up2date`, `yum`, or `yast`) in the same way as you would use them on common non-Virtuozzo systems. This is due to the fact that the Virtuozzo installation program modifies the settings of these updaters in such a way that the kernel and other packages vital for Virtuozzo functioning do not get updated, unlike all the other operating system packages. The Hardware Node administrator should regularly use these updaters without overriding their default behavior, which ensures that the non-Virtuozzo-specific part of the operating system has all the latest fixes (including security patches) installed and that Virtuozzo-specific packages are not erroneously updated by native updaters. If a security patch or other fix is issued for the mainstream Linux kernel or any other package that has been modified for Virtuozzo needs, the kernel/package is instantly rebuilt by SWsoft with this security patch and becomes accessible on the Virtuozzo update site (see the **Updating Virtuozzo Software** section below).

There follows a description of peculiarities of various native updaters and their integration with Virtuozzo.

Using up2date

The `up2date` updater can be used on such host operating systems as Red Hat 9, Red Hat Enterprise Linux 3, Red Hat Enterprise Linux 4, CentOS 3, CentOS 4, Fedora Core 1, Fedora Core 2, and Fedora Core 4.

If Virtuozzo is installed on the computer, the `/etc/sysconfig/rhn/up2date` configuration file is modified in such a way so as to let all the packages necessary for Virtuozzo functioning remain intact on the Node during the work of `up2date`.

So, here are the pitfalls you should avoid while using the `up2date` updater:

- Do not modify the value of the following parameters in the `/etc/sysconfig/rhn/up2date` configuration file:
 - `pkgSkipList`;
 - `removeSkipList`;
 - `forceInstall` (the right value is 0);
- Do not pass the `-f` or `--force` option to `up2date`.

Using yum

The `yum` updater can be used on such host operating systems as CentOS 3, CentOS 4, Fedora Core 1, Fedora Core 2, and Fedora Core 4.

If Virtuozzo is installed on the computer, the `/etc/yum.conf` configuration file is modified in such a way so as to exclude all the packages necessary for Virtuozzo functioning from the update list. The parameter the value of which you should avoid modifying in this configuration file is `exclude`, which is located in the `[main]` section. You should not use other configuration files than the default one, either.

Using yast

The `yast` updater can be used on the SUSE Linux Enterprise Server 9 host operating system. The default behavior of this updater does not update any packages that are not signed by SUSE. All the packages on a SLES 9 distribution that have been replaced with analogous ones by SWsoft (including the kernel) do not have the SUSE signature and thus are excluded from the default update list. However, it is very easy to inadvertently override this default behavior (a casual mouse-click opposite an SWsoft-rebuilt package could be enough), so you should be extra careful with `yast`.

Updating Virtuozzo Software

Beginning with Virtuozzo 2.6.1, Virtuozzo installations can be updated with the help of the `vzup2date` utility, which connects to the Virtuozzo update server and updates the following components on the Hardware Node:

- the kernel;
- the Linux packages copyrighted by third parties (by the OS vendor, for example) but built by SWsoft for compatibility with Virtuozzo. Such packages are usually rebuilt by SWsoft and put on the Virtuozzo update server after a security or other important hotfix is issued by the third party.
- the Virtuozzo packages copyrighted and built by SWsoft, Inc.;
- the standard Virtuozzo templates (unlike new EZ templates, which do not require updates) installed on the Hardware Node.

Full information on the `vzup2date` utility is given in the [Updating Virtuozzo with vzup2date](#) section of the Virtuozzo Installation Guide and in the `vzup2date` section of the Virtuozzo Reference Guide.

Updating Software Inside Virtual Private Servers

The majority of software packages are added to Virtual Private Servers by means of the Virtuozzo templates technology. Depending on the type of templates the given VPS uses, the update process is different:

- If the VPS uses the standard templates technology, then the first thing to do is updating the corresponding templates (see page 191). After that, the updated versions of the templates can be added to the VPSs that use them as is described in the [Adding Templates and Template Updates to VPSs](#) section (on page 81).
- If the VPS uses the EZ templates technology, it is updated by means of the `vzpkg update <vpsid>` command. For more information on this command, see the [Updating EZ Templates](#) section in the Virtuozzo Installation Guide.

Note: If a certain software package is manually installed directly inside a VPS, i.e. without resorting to the Virtuozzo templates technology, the process of its updating should also be manual (unless this package is part of an EZ template repository, in which case you can continue using the `vzpkg update` command).

CHAPTER 11

Customizing VZCC/VZPP

This chapter contains instructions for experienced administrators on the way to customize Virtuozzo Control Center and Virtuozzo Power Panels by changing the VZCC/VZPP layout and appearance and modifying certain VZCC/VZPP settings. Besides, it explains how to set up the demo mode to allow your potential customers to manage real Virtual Private Servers by means of VZCC/VZPP.

In This Chapter

Overview	193
Common Tasks	194
VZCC/VZPP Architecture	199
Customizing Individual Pages.....	201
Customizing VZCC/VZPP Menu	206
Customizing VZCC/VZPP Layout	208
Adding Skins to VZCC/VZPP	208
Setting Up VZCC/VZPP Demo Mode for Potential Customers	210
Importing Plesk Language Packs Into VZPP	219

Overview

The Hardware Node administrator may customize Virtuozzo Control Center and Virtuozzo Power Panels by altering the VZCC/VZPP layout and appearance (e.g. including additional links on the menu) as well as by modifying certain VZCC/VZPP settings. This chapter begins with the step-by-step descriptions of the most common tasks you are likely to perform:

- adding your own links to the VZCC/VZPP menu (see page 194);
- setting automatic logout time (see page 196);
- setting the maximal number of allowed VPS backups (see page 197);
- adjusting the appearance of traffic accounting bars and tables (see page 198).

If you are looking for other ways to customize VZCC/VZPP, you are recommended to read on and peruse the **VZCC/VZPP Architecture** section (on page 199) as well as the consequent sections devoted to the following tasks:

- modifying an existing VZCC/VZPP page (by the example of the VZPP services page) (see page 201);
- adding a new VZCC/VZPP page (by the example of adding a new dashboard to the VZPP menu) (see page 203);
- customizing the VZCC/VZPP menu (see page 206);
- customizing the VZCC/VZPP general layouts (see page 208);
- changing VZCC/VZPP graphical skins (see page 208).

The **Setting Up VZCC/VZPP Demo Mode for Potential Customers** section (on page 210) describes the way to set up the demo mode for Virtuozzo Control Center and Virtuozzo Power Panels to allow your potential customers to manage real Virtual Private Servers thru VZCC and VZPP.

Certain VZCC/VZPP parameters that can also be customized are defined in the VZCC/VZPP configuration file (for detailed information see the **VZCC/VZPP Configuration File** section in the **Configuring Virtuozzo** chapter of the **Virtuozzo Reference Guide**).

When customizing VZCC/VZPP, you are supposed to work with XML and XSL files located either inside the Service VPS or on the Hardware Node itself. You can access the Service VPS in one of the following ways:

- 1** Log in to the Service VPS via `ssh` with the `root` account. You should make sure that you have enabled the `root` account for the Service VPS; if not, turn to **Setting Root Password for VPS** (on page 46) for instructions on how to do it. Then use any text editor like `vi` for modifying the necessary files.
- 2** Log in to the Hardware Node where the Service VPS is hosted and go to the `/vz/root/1/` directory, which is the root directory for the Service VPS. Then, again, any text editor can be used for modifying the files.
- 3** Use the VZMC file manager for modifying the necessary text files inside the Service VPS, i.e. download the corresponding files to your local computer, edit them, and upload the files again to the Service VPS.

Common Tasks

Adding VZCC/VZPP Links

Adding your own links to the VZCC or VZPP menus is performed by editing the `/etc/vzcp/cc/menu.xml` or `/etc/vzcp/pp/menu.xml` files inside the Service VPS, respectively. Let us assume that you wish to provide VZPP with a link to the page on the provider's website where the VPS administrator is able to buy additional resources for their VPS. To this effect, you should open the `/etc/vzcp/pp/menu.xml` file for viewing and editing. For example, you can issue the following command on the Node to edit the VZPP menu:

```
# vi /vz/root/1/etc/vzcp/pp/menu.xml
```

Note: To add a new link to the VZCC menu, you should open the `/etc/vzcp/cc/menu.xml` file inside the Service VPS and edit it in the same way as is described below for the VZPP menu.

As you see, there are four `<submenu>` elements in this file. Each `<submenu>` element is opened with the `<submenu list_of_attributes>` tag and closed with the `</submenu>` tag. In the browser, each `<submenu>` element corresponds to one of the four groups of links on the VZPP menu. Let us add a link to the first group of links called VPS Management:

- 1 In the `/etc/vzcp/pp/menu.xml` file, locate the `<submenu>` element having `name="VPS Management"` as one of its attributes. Between the `<submenu>` opening tag and the `</submenu>` closing tag, there are a number of `<item>` elements, also having their corresponding opening and closing tags. Each `<item>` element corresponds to a link on the VZPP menu.
- 2 Create a copy of one of the `<item>` elements (i.e. all the text between the `<item>` opening tag and the `</item>` closing tag, including the tags proper) and place it just before the `</submenu>` closing tag. This will be a basis for your link.
- 3 Edit this `<item>` element in the following way:
 - Enter the path to the page where the VPS administrator can buy more resources for their VPS as the value of the `href` attribute.
 - Enter the ID of the icon file to be shown to the left of the link as the value of the `icon` attribute. The file should be located in the `/var/vzcp/static/vz/skins/<skin_name>/icons` directory inside the Service VPS and have the following name pattern: `<id>_16.gif`.

Note: There are a number of subdirectories in the `/var/vzcp/static/vz/skins/` directory. One subdirectory corresponds to one VZPP skin defining such VZPP elements as its general layout, icons and images, the color palette, etc. Therefore, to allow your new icon to be displayed in VZPP when using any of the existing skins, you should add the icon file to each subdirectory.

- Between the `<title>` and `</title>` tags, enter the text to be shown as the new link in the browser.

- Remove the remaining attributes of the `<item>` element and the existing `<screen>` elements.

In the end, your new `<item>` element may look like below:

```
<item href="http://www.domain.com/res.cfm" icon="res_16"
target="_blank">
  <title>Buy resources</title>
</item>
```

Note: The `target="_blank"` attribute tells the browser to open the link in a new browser window. Omitting this attribute will open the link in the same browser window.

The newly created link will be available for all VPS administrators having access to the `<submenu>` element where your new `<item>` element is included. However, you may wish to restrict access to the link. The `cred` attribute in the `<item>` element allows you to prevent the link from being displayed on the VZPP menu. To this effect, you should enter a random ID as the value of the `cred` attribute and make sure that this ID is not specified in the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS as one of the existing XML elements included in the `<access>` pattern element.

To make the link available on the VZPP menu again, you can choose one of the following ways:

- Remove the `cred` attribute from the `<item>` element in the `/etc/vzcp/pp/menu.xml` file inside the Service VPS;
- Specify the `cred` attribute in the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS as an XML element (for example, `<buy_resources/>`). The name of the element should coincide with the ID of the `cred` attribute in the `/etc/vzcp/pp/menu.xml` file, and the element itself should be placed between the following pattern elements:

```
<auth>
  <type>VEUser</type>
  <user>root</user>
  <creds>
    <management>
```

and

```
    </management>
  </creds>
</auth>
```

In this case the `root` user will gain access to the created link, i.e. this link will be displayed on the VZPP menu under the VPS Management group when the `root` user is logged in to the Virtual Private Server.

Note: If you are creating a new VZCC link, you should look for the `<type>SVEUser</type>` string instead of `<type>VEUser</type>` in the `/etc/vzcp/vzcpcon.conf` file and place the corresponding XML element between the subsequent `<management>` opening tag and the `</management>` closing tag.

While working with the `cred` attribute, you should also keep in mind the following:

- After you have defined a new element in the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS, you should restart the `vzcp` service by executing the `/etc/init.d/vzcp restart` command and log out of and log in to VZPP afresh for the new settings to take effect.

- b** Do not use spaces in the value of the `cred` attribute.

To learn more about the VZCC/VZPP menu customization, turn to the **Customizing VZCC/VZPP Menu** section (see page 206).

Setting Auto Logout Time

If a VZCC/VZPP user has been inactive for a certain time after logging in to VZCC/VZPP, the system automatically logs them out, and the continuation of working with VZCC/VZPP is possible only after logging in anew. You may customize the period of inactivity before an automatic logging out by editing the `vzagent` configuration files (`/etc/sysconfig/vzagent/vzagent.conf`). One file is located inside the Hardware Node, and the other one - inside the Service VPS. Open each file for editing and find the following fragment of code inside the file:

```
<ticketm>
  <configuration>
    <ticket_expiration>10800</ticket_expiration>
  </configuration>
</ticketm>
```

The value of the `<ticket_expiration>` element denotes this period of inactivity in seconds (3 hours in the example above). Edit this value as required and save the file. After you have appropriately edited both files, restart `vzagent` by executing the following command on the Hardware Node:

```
# vzagent_ctl restart
```

A simpler way of editing the `vzagent` configuration is by using VZMC:

- 1** Right-click the Hardware Node name in the VZMC left pane and select **Tasks --> Manage VZAgent Configuration** on the context menu.
- 2** In the **VZAgent Configuration** window, expand the `ticketm` key and select the `configuration` subkey.
- 3** Double-click the `ticket_expiration` parameter in the right pane:

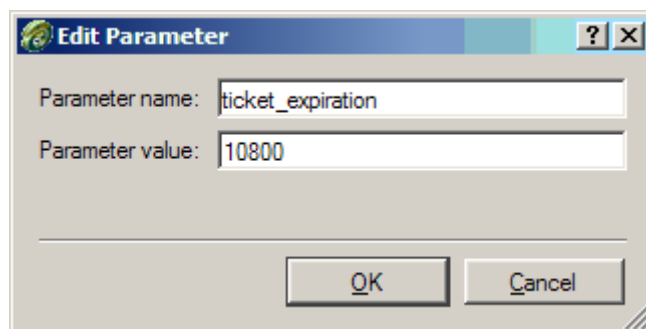


Figure 46: VZMC - Setting VZCP Auto Logout Time

- 4** In the **Edit Parameter** window, enter the value you want in the **Parameter value** field.
- 5** Press **OK**.

The change will take effect immediately. In VZMC, you need to edit the value only once, and you do not need to restart `vzagent` manually.

Note: The `ticket_expiration` parameter set in the `/etc/sysconfig/vzagent/vzagent.conf` file defines the logout time for both VZCC and VZPP.

Setting Maximal Backups Number

Service VPS users and VPS administrators may be able to back up their Virtual Private Servers by means of VZCC and VZPP, respectively. VPS backups are stored on the Hardware Node and may take up a vast amount of disk space. That is why the Hardware Node administrator may wish to limit the number of VPS backups Service VPS users and VPS administrators are allowed to create. This number can be customized by editing the `vzagent` configuration files (`/etc/sysconfig/vzagent/vzagent.conf`). One file is located inside the Hardware Node, and the other one - inside the Service VPS. Open each file for editing and find the following fragment of code inside the file:

```
<backm>
  <configuration>
    a number of lines...
    <pe_backups_limit>1</pe_backups_limit>
    a number of lines...
  </configuration>
</backm>
```

The value of the `<pe_backups_limit>` element denotes this number of backups (1 is the default value). Edit this value as required and save the file. After you have appropriately edited both files, restart `vzagent` by executing the following command on the Hardware Node:

```
# vzagent_ctl restart
```

A simpler way of editing the `vzagent` configuration is by using VZMC:

- 1 Right-click the Hardware Node name in the VZMC left pane and select **Tasks --> Manage VZAgent Configuration** on the context menu.
- 2 In the **VZAgent Configuration** window, expand the `backm` key and select the `configuration` subkey.
- 3 Double-click the `pe_backups_limit` parameter in the right pane.
- 4 In the **Edit Parameter** window, enter the value you want in the **Parameter value** field.
- 5 Press OK.

The change will take effect immediately. In VZMC, you need to edit the value only once, and you do not need to restart `vzagent` manually.

Note: The limits set on the number of VPS backups by editing the `vzagent` configuration files concern only VPS administrators and Service VPS users. There are no restrictions for a Hardware Node administrator (i.e. for the `vzagent0` user), who is allowed to create as many VPS backups as s/he wants to.

Adjusting Traffic Representations

VZCC and VZPP provide the opportunity to view the network traffic and bandwidth statistics for a particular VPS on the **Traffic Statistics** and **Resources** pages, respectively.

On the **Resources** page, the information on the network bandwidth is presented in the form of the **Bandwidth** column where a special bar graphically shows what percent of the network bandwidth has been used for outgoing traffic in case there are restrictions on the outgoing traffic for the given Virtual Private Server. The period the bandwidth statistics will be showed for can be defined in VZMC by selecting **Monitor --> Log Setup** under the corresponding Hardware Node name, clicking **Network Traffic** under the **Virtual Private Servers** group in the right pane, and specifying the needed period in the **New logging period** field. By default, this time period is set to 5 minutes.

The information on the amount of network traffic is summarized in the **Traffic for selected period** table on the VZCC/VZPP **Traffic Statistics** page and displays how much traffic has gone out of the VPS during the time span defined. The time period for which you wish to see your traffic statistics is specified in the fields provided on the **Traffic Statistics** page.

By default, the statistics on the total bandwidth and traffic is shown, irrespective of Virtuozzo network classes (see page 132). However, you may wish to show the statistics separately for each class. To this effect, you should modify the `/etc/vzcp/netcounters.xml` file inside the Service VPS. This file is in the XML format and consists of a number of `<counter>` elements responsible for displaying a separate network statistics table per network class on the **Traffic Statistics** page and a separate graphical bar per network class on the **Resources** page. Each `<counter>` element should have one and only one `<name>` subelement defining how this table and bar should be called, and any number of `<class>` subelements defining the network classes for which the statistics will be shown. If there are two or more `<class>` subelements, the statistics is summed up for all of them. The network classes themselves are defined in the `/etc/sysconfig/vz-scripts/networks_classes` file on the Hardware Node.

VZCC/VZPP Architecture

The way VZCC/VZPP pages are organized and presented in a browser is determined by the corresponding XSL tables installed inside the Service VPS. It means that the layout and appearance of VZCC/VZPP pages are not hard-coded anywhere but may be customized as needed by editing these XSL files to create an individual look and feel for your VZCC/VZPP. Thus, the following things may be done:

- Customizing individual VZCC/VZPP pages, including dashboards;
- Customizing the VZCC/VZPP menu;
- Customizing the VZCC/VZPP general layout;
- Creating skins for VZCC/VZPP (whole sets of CSS and image files as well as some other parameters) and switching among various skins.

All these tasks will be explained and covered in the following sections. In a general case, it is enough to possess an adequate HTML knowledge to customize the pages. However, it is also very desirable to know the basics of XML syntax. An expertise in the XSL language will give you the complete control over the VZCC/VZPP presentation.

Before reading on, it is important to understand in what way the XSL stylesheets take part in the process of making up HTML pages displayed in the VPS administrator's browser. The scheme below illustrates the actions performed inside the Service VPS between receiving a request from the browser and sending back the resulting HTML page to it:

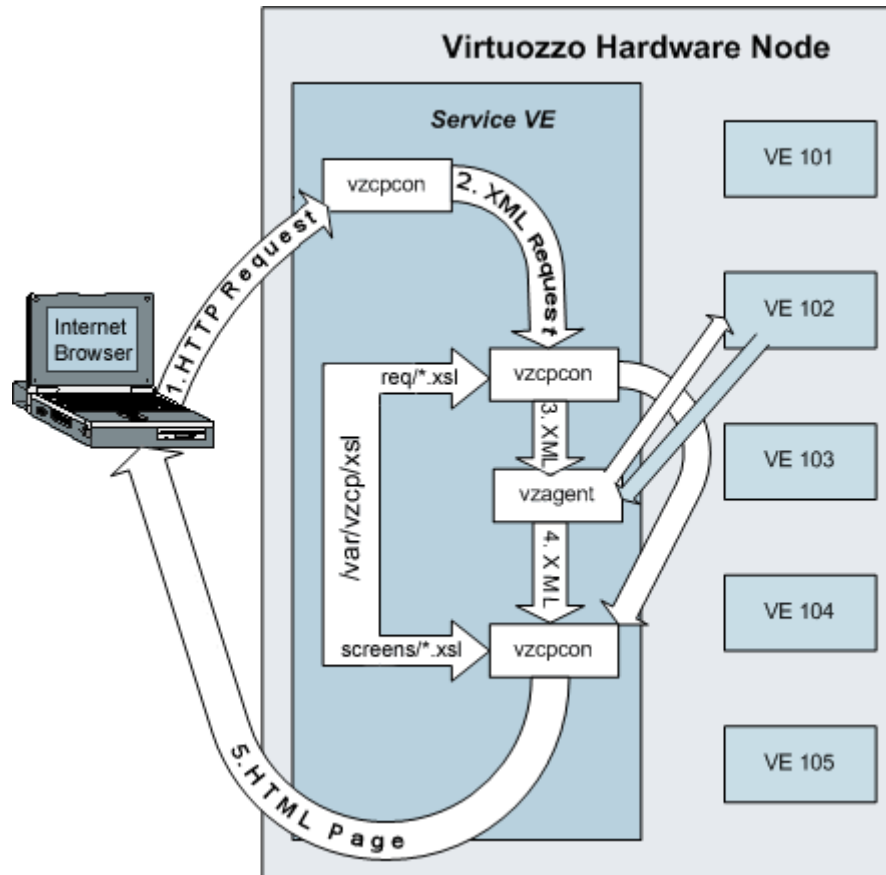


Figure 47: VZCC/VZPP Transformations

Now each action is treated in more detail:

- 1 The administrator follows, say, the `http://<Service_VPS_IP_address:TCP_port>/cp/pwd` link in his browser (or enters this link in the browser address line, for that matter). This page is used to change the `root` password in the current VPS. The administrator may use the IP address not of the Service VPS, but of a particular VPS s/he is managing. It is not of much importance here, since any such request is anyway redirected to the Service VPS.
- 2 The `vzcpcon` process inside the Service VPS handles the HTTP request from the browser and converts it to an XML request (the contents of the HTTP request presented in the XML format).
- 3 The `vzcpcon` process converts the created XML request to another XML format suitable for parsing by `vzagent`. To perform this action, `vzcpcon` uses the corresponding `*.xsl` file located in the `/var/vzcp/xsl/req` directory inside the Service VPS (or, the `/vz/root/1/var/vzcp/xsl/req` directory on the Hardware Node). In our case, the `pwd.xsl` file is used.

Note: If the request does not need a response from the VPS being managed (for example, the request is for displaying a dashboard common for all the VPSs), and therefore, `vzagent` is not involved, `vzcpcon` proceeds directly to Step 5. This is shown as a side arrow on the scheme above.

- 4 `vzagent` handles the request by connecting to the corresponding VPS and makes up a response to `vzcpcon`. This response is also in the XML format, since XML is the primary language of `vzagent`.
- 5 The `vzcpcon` process merges this XML response with the corresponding `*.xsl` file located in the `/var/vzcp/xsl/screens` directory inside the Service VPS (or the `/vz/root/1/var/vzcp/xsl/screens` directory on the Hardware Node). In our case, it will be the `pwd.xsl` file. Any such `xsl` file can be regarded as a set of templates instructing `vzcpcon` of the way to convert the XML response from `vzagent` to a valid HTML page. The resulting HTML page is sent back to the Internet browser and displayed in the administrator's window.

Thus, we see that it is modifying HTML templates inside the `*.xsl` files that tells upon the way the VZCC/VZPP pages are generated. To learn more about this process and to get to know certain nuances of customizing VZCC/VZPP, see the following subsections.

Customizing Individual Pages

You may alter the appearance of a basic VZCC/VZPP page, or you may add your own pages to VZCC/VZPP. These tasks are described below.

Modifying Existing Page

To modify an existing VZCC/VZPP page, you should first determine what `xsl` file is responsible for displaying this or that page. Let us assume you wish to reorganize the **Services** table in VZPP:

Note: To modify a VZCC page, you should perform the same operations that are described below for customizing a VZPP page.

- 1 Log in to VZPP and click the **System Services** link on the left menu. If you are using the framed layout for VZPP, you may determine the URL of the page, for example, by placing the mouse on the link and looking at the status bar of your browser. In our case, it should read `.../vz/cp/services`. This means that this page is displayed with the help of the `services.xsl` file in the `/var/vzcp/xsl/screens` directory inside the Service VPS.

Note: The `/var/vzcp/xsl/screens` directory inside the Service VPS contains the `*.xsl` files common for both VZCC and VZPP. However, there are some `*.xsl` files used for creating VZCC pages only. These files are located in the `/var/vzcp/xsl/screens/cc` directory inside the Service VPS.

- 2 Log in to the Service VPS with the root account and open the `/var/vzcp/xsl/screens/services.xsl` file for viewing and editing. You should make sure that you have enabled the root account for the Service VPS; if not, turn to **Setting Root Password for VPS** (on page 46) for instructions on how to do it. If you are logged on to the Hardware Node where the Service VPS is hosted, but not to the Service VPS itself, you may as well work with the `/vz/root/1/var/vzcp/xsl/screens/services.xsl` file. Still, another possibility for working with text files is provided by the VZMC file manager feature.
- 3 To unravel the way this (and any) xsl file forms the HTML page, you should search for the following string in the file:

```
<xsl:template name="screen">
```

This element marks the beginning of generating an HTML page, and, if you know the XSL language, you may learn what auxiliary templates it calls (located either in the `services.xsl` file or in other files enumerated as attributes in the `<xsl:include/>` elements at the end of the file) and modify these templates as needed. If XSL is not your forte, you should search for the HTML code in the `services.xsl` file resembling the table on the **System Services** page. For example, you may guess that the following code is responsible for displaying the header row in the table:

```
<xsl:with-param name="browse">
  <ui:browse-setup id="services" item-url="/service-win"
    url="/services-win" item-url-tip="service_screen"/>
  <ui:cols>
    <ui:column id="name" hidden="yes" for-url="yes" for-select="yes"
      encode="no"/>
    <ui:column id="display-name" nav-type="id"
      title="{sw:text('Name')}}" filter="auto" for-select="no"
      for-url="no"/>
    <ui:column id="status" sorting="yes" align="center"
      title="{sw:text('Status')}}" />
    <ui:column id="startup-type" sorting="yes" align="center"
      title="{sw:text('Startup Type')}}" />
    <ui:column id="status-hidden" title="{sw:text('Status')}}"
      filter="select" hidden="yes">
      <ui:filter-option value="Stopped" title="{sw:text('Stopped')}}" />
      <ui:filter-option value="Running" title="{sw:text('Running')}}" />
    </ui:column>
    <ui:column id="startup-type-hidden" title="{sw:text('Startup
Type')}}"
      filter="select" hidden="yes">
      <ui:filter-option value="Automatic"
        title="{sw:text('Automatic')}}" />
      <ui:filter-option value="Manual" title="{sw:text('Manual')}}" />
      <ui:filter-option value="Disabled" title="{sw:text('Disabled')}}" />
    </ui:column>
    <ui:column id="logon-as" filter="auto" sorting="yes" align="center"
      title="{sw:text('Logon as')}}" />
  </ui:cols>
</xsl:with-param>
```

- 4 Modify the HTML code as needed. Mind that this code should comply with the XHTML requirements, i.e. you should close all orphan elements (e.g. `
` instead of `
`) and do not use cross-nested elements (e.g. `<i>text</i>` is correct, whereas `<i>text</i>` is incorrect). If you are unsure whether your modifications comply with the XHTML requirements, you may use third-party software, for example, XMLSPY (www.altova.com) to check whether the resulting file is well-formed.

For example, you may exchange the places of the following fragments of code:

```
<ui:column id="startup-type" sorting="yes" align="center"
  title="{sw:text('Startup Type')}" />
```

and

```
<ui:column id="status-hidden" title="{sw:text('Status')}"
  filter="select" hidden="yes">
  <ui:filter-option value="Stopped" title="{sw:text('Stopped')}" />
  <ui:filter-option value="Running" title="{sw:text('Running')}" />
</ui:column>
```

After that, the **Status** and **Startup Type** columns will be exchanged in the **Services** table on the System Services page. Refresh your VZPP page to see the results.

Note: The VZPP Services page reflects not all the services inside the VPS, but only those services which are also present in the `/etc/vzcp/services.xml` file inside the Service VPS. To be able to display additional services, you should add the corresponding `<service/>` elements to this file.

Adding New Page

New VZCC/VZPP pages are created by adding new `*.xsl` files to the `/var/vzcp/xsl/screens` directory inside the Service VPS. Since these `xsl` files are written specifically for the XML generated by `vzagent`, it is recommended not to create new files from scratch but use existing files as a starting point. In this subsection we will consider a situation when you want to add a new dashboard to your VZPP, say, for housing external links related to your business. So, here is what you should do (we suppose that you are working inside the Service VPS):

Note: The `/var/vzcp/xsl/screens` directory inside the Service VPS contains the `*.xsl` files common for both VZCC and VZPP. However, there are some `*.xsl` files used for creating VZCC pages only. These files are located in the `/var/vzcp/xsl/screens/cc` directory inside the Service VPS.

- Decide on any existing VZPP dashboard file and make a copy of it in the `/var/vzcp/xsl/screens` directory:

```
# cd /var/vzcp/xsl/screens
# cp vps.xsl my_dash.xsl
```

- Open the newly-created file for editing:

```
# vi my_dash.xsl
```

- Find the following line in the file:

```
<ui:screen id="07.03.00.00" title="{sw:text('VPS Management')}"
  icon="vps_management" cred="/vps/management">
```

and change the value of the following attributes:

- `id`. When defining the `id` values for your own VZPP pages, you should leave the first dyad intact (it identifies VZPP as a module), change the second one to a large number (to identify your personal screens), and have the last two ones at your choice (they would identify different subscreens within the group of your personal screens). Thus, you may define the following ID for your VZPP page:

```
<ui:screen id="07.99.01.00" ...>
```

- **title.** Change the value of the attribute to what you wish to be reflected as the title of your dashboard on the VZPP page.
- **icon.** Change the value of the attribute to point to the icon file you wish to be displayed to the left of the dashboard title. The icon file itself should be placed to the `/var/vzcp/static/vz/skins/<skin_name>/icons` folder with the following name mask: `<id>_32.gif`. The `id` value of this mask should be entered as the value of the select attribute instead of the `vps_management` string.

Note: There are a number of subdirectories in the `/var/vzcp/static/vz/skins/` directory. One subdirectory corresponds to one VZPP skin defining such VZPP elements as its general layout, icons and images, and the color palette. Therefore, to allow your new icon to be displayed when using any of the existing skins, you should place the icon file in each subdirectory.

- **cred.** Optional. Change the ID of the `cred` attribute responsible for granting VPS administrators access to this link on the VZPP menu. This ID should be specified in the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS between the `<auth>` and `</auth>` tags. Detailed information on the `cred` attribute is provided in the [Adding VZCC/VZPP Links](#) subsection (see page 194).
- Inasmuch as there is no need in connecting to any VPS for displaying any dashboard page, and so `vzagent` is not involved in the process, `vzcpcon` uses not the XML response from `vzagent` to build an HTML page, but the static `/etc/vzcp/pp/dashboard.xml` file. So, you further need to define your dashboard in this file. Open it for editing:

```
# vi /etc/vzcp/pp/dashboard.xml
```

Note: While adding a new VZCC page, you should work with the `/etc/vzcp/cc/dashboard.xml` file.

There are four `<dashboard>` elements defined in this file, including the top dashboard providing links to the three others.

- Copy one of the children `<dashboard>` elements inside the file to provide a basis for defining your own dashboard, for example, the **Services** dashboard:

```
<!-- VPS Services dashboard -->
<dashboard id="07.04.01.00" name="VPS Services" cred="/vps/services">
  <item href="$baseurl/services" title="System Services"
    icon="services" tip="srv_screen" cred="system_services"
    platform="linux">
    <text>On this screen you may manage various services
      in this VPS.</text>
  </item>
  <item href="$baseurl/proc" title="System Processes"
    icon="processes"
    tip="proc_screen" cred="system_processes">
    <text>On this screen you may check processes currently
      running in your VPS and send them system signals.</text>
  </item>
  ...
</dashboard>
```

- Now edit this dashboard definition to reflect your own template:
 - Edit the commentary describing your dashboard;

- Edit the value of the `id` attribute of the `<dashboard>` element to reflect the screen ID you have specified in the `my_dash.xml` file;
- Edit the value of the `name` attribute of the `<dashboard>` element to provide a good description to your dashboard. There are no restrictions as to what this name should be;
- Create as many `<item>` elements as the number of links on your dashboard;
- Edit the value of the `href` attribute of each `<item>` element. This will be the value of the `href` attribute of the `<a>` element in the resulting HTML. You may specify an absolute path here as well.
- Edit the value of the `title` attribute of each `<item>` element. This will be the text between the `<a>` and `` tags in the resulting HTML.
- Edit the value of the `icon` attribute of each `<item>` element to point to the icon you wish to be displayed to the left of the title. See above for details on how this value should be specified.
- Edit the value of the `tip` attribute of each `<item>` element to display the brief description of the page currently opened in VZPP. The value of the `tip` attribute should be specified as the `id` attribute of the `<tip>` element in the `/etc/vzcp/tips.xml` file inside the Service VPS.
- Edit the value of the `cred` attribute of each `<item>` element, if any, to grant VPS administrator access to the VZPP page. Detailed information on the `cred` attribute is provided in the **Adding VZCC/VZPP Links** subsection (see page 194).
- Enter `linux` as the value of the `platform` attribute of each `<item>` element if you wish the link defined by this `<item>` element to be available on Linux Hardware Nodes only. Otherwise, remove this attribute from the `<item>` element.
- You may additionally define the `target` attribute for any `<item>` element that would be copied as the `target` attribute for the `<a>` element in the resulting HTML.
- Edit the text between the `<text>` and `</text>` tags for each `<item>` element. This will be the description of your link on the HTML page.
- At last you have to create an additional `<item>` element for the top dashboard. For example, you may define it as follows:

```
<item href="$baseurl/my_link" title="External Links"
icon="external_links">
  <text>
    Various external links gathered for your convenience.
  </text>
</item>
```

That is almost all about creating a new dashboard of yours. You may now access it by typing

`https://<Service_VPS_IP_address_or_hostname>:<TCP_port>/cp/my_dash`

into the address line of your browser or by following the proper link from the top VZPP dashboard. However, after creating your new dashboard, you may naturally want to provide a link to it in the left VZPP menu. Please consult the **Adding VZCC/VZPP Links** subsection (see page 194) to know how to add your own links to the VZPP menu.

Customizing VZCC/VZPP Menu

Inasmuch as there is no need in connecting to any VPS for displaying the VZCC/VZPP menu on any page, and so `vzagent` is not involved in the process, `vzcpcon` uses not the XML response from `vzagent` to build the menu, but the static `/etc/vzcp/cc/menu.xml` and `/etc/vzcp/pp/menu.xml` files inside the Service VPS, respectively. So, you may define your additional links and submenus in these particular files or change the properties of the existing ones. Detailed information on how to add your own links and submenus to the VZCC/VZPP menu is provided in the **Adding VZCC/VZPP Links** subsection (see page 194). The given subsection familiarizes you with the way to modify the existing links and submenus on the VZCC/VZPP menus only.

Let us assume you wish to edit the VZCC menu. To this effect, you should open the `/etc/vzcp/cc/menu.xml` file for editing:

```
# vi /etc/vzcp/cc/menu.xml
```

As you probably know from the **Adding VZCC/VZPP Links** subsection (see page 194), there are three `<submenu>` elements defined in this file by default. One `<submenu>` element constitutes one group on the VZCC menu providing links to the existing dashboards and grouping the links to individual VZCC pages presented by the `<item>` elements inside each `<submenu>` element. The properties of all `<submenu>` and `<item>` elements are defined by the values of their attributes and/or their subelements; so you should simply change these values to customize the VZCC menu properties. The table below describes the meaning of each attribute/subelement in the `/etc/vzcp/cc/menu.xml` file:

Attribute/subelement	Description
<code>name</code>	The name of the dashboard link as it will be displayed on the menu.
<code>href</code>	The path the link references. This may be either an absolute or relative path. The <code>\$baseurl</code> variable stands for <code>https://<VPS_IP_address>:TCP_port/vz/cp</code> corresponding to <code>/var/vzcp/xsl/screens</code> inside the Service VPS. If the path specifies an xsl file in this directory, its extension should be omitted.
<code>cred</code>	The ID of the <code>cred</code> attribute responsible for granting a Service VPS user access to this or that link on the VZCC menu. This ID should be specified in the <code>/etc/vzcp/vzcpcon.conf</code> file inside the Service VPS between the <code><auth></code> and <code></auth></code> tags. Detailed information on the <code>cred</code> attribute is provided in the Adding VZCC/VZPP Links subsection (see page 194).
<code>tip</code>	The value of the <code>id</code> attribute of the <code><tip></code> element in the <code>/etc/vzcp/tips.xml</code> file inside the Service VPS. The contents of the corresponding <code><tip></code> element is the description of the link the Service VPS user sees when moving the mouse over the link.
<code>screen-id</code>	The ID of the screen the link refers to. This attribute has no special meaning apart from highlighting the link when you are on the corresponding VZCC page.
<code>type</code>	Defines the link area where the contents of the <code><tip></code> element will be shown when moving the mouse over the corresponding element on the VZCC interface.

icon	The ID of the icon file to be shown to the left of the link. The file should be located in the <code>/var/vzcp/static/vz/skins/<skin_name>/icons</code> directory inside the Service VPS and have the following name pattern: <code><id>_16.gif</code> .
id	The ID of the screen the link refers to. This attribute has no special meaning apart from highlighting the link when you are on the corresponding VZCC page.
target	An optional attribute of the <code><item></code> element acting as the target attribute of the HTML <code><a></code> element. For example, specifying <code>target="_blank"</code> will tell the browser to open the corresponding link in a new window.
svps	Indicates if the operation defined by the <code><item></code> element where this attribute is specified can be performed in the Service VPS context. To prevent the operation to be applied to the Service VPS, you should set the value of this attribute to <code>deny</code> . Omitting the attribute or specifying another value in this attribute will allow you to perform the operation in the Service VPS context.
platform	If specified, indicates the platform (Hardware Node) where the link defined by the <code><item></code> element containing this attribute will be displayed on the VZCC menu. The values that can be set in this attribute are the following: <ul style="list-style-type: none"> win32: the link will be shown for all Hardware Nodes running the Windows Server 2003 OS; linux: the link will be displayed for all Hardware Nodes running the Linux OS (Red Hat 9, Fedora Core 3, etc.).
cond	Is used to additionally check if the link should be displayed on the VZCC menu.
<title>	The name of the link under the dashboard link as it will be displayed on the menu to the right of its icon.
<screen>	Attaches screens whose identifiers are listed as the value of the <code>id</code> attribute.
<custom_script>	Indicates what Java script should be executed while clicking the link on the VZCC menu.
<template>	Is used to check the version of the Confixx template and, depending on the version, to decide whether the Confixx link should be shown on the VZCC menu or will be unavailable for users.

Note: The aforementioned attributes and subelements can be specified in both files `/etc/vzcp/pp/menu.xml` and `/etc/vzcp/cc/menu.xml`, i.e. any of these elements can be used for building both the VZPP menu and the VZCC menu.

The appearance proper of the VZCC menu is controlled by the `/var/vzcp/xsl/lib/nav_menu.xsl` file inside the Service VPS. You may edit the HTML code in this file to create a different look for the menu. This procedure is covered in the Customizing Individual Pages subsection above.

Customizing VZCC/VZPP Layout

If you simply want to define a kind of layout (framed, non-framed, vzcc, etc.) to be used for building VZCC/VZPP pages, you should turn to the **Adding Skins to VZCC/VZPP** subsection for the instructions.

If you want to perform a more intricate customization, you should modify the code inside one of the `.xsl` files in the Service VPS `/var/vzcp/xsl/layout/` directory. The `.xsl` files from this directory are used to transform the XML response from `vzagent` - a set of `.xml` files containing all the needed information on your pages, dashboards, menus, etc. in Virtuozzo Control Center and Virtuozzo Power Panels - to the HTML pages as they are displayed on your screen. In other words, these files are responsible for forming the VZCC/VZPP layout, although to a variable extent. For example, you can edit the `plain.xsl` or `framed.xsl` files in the `/var/vzcp/xsl/layout/` directory, which build up the body of the HTML page, to define the general VZCC/VZPP layout (the appearance and relative positions of the VZCC/VZPP menu, title bar, etc.). In its turn, the complete HTML page as such, i.e. beginning with its essential `<html><head>` elements and so on, is made up by the `basic.xsl` file located in this directory; so, you may modify this file to customize the VZCC/VZPP appearance.

Note: To edit the `.xsl` files in the `/var/vzcp/xsl/layout/` directory, you should have sound knowledge of Extensible Stylesheet Language Transformations (XSLT).

After you have edited and saved any layout `.xsl` file, it is necessary to restart `vzcp` for the changes to take effect:

```
# /etc/init.d/vzcp restart
Stopping vzcp: [ OK ]
Starting vzcp: [ OK ]
```

Adding Skins to VZCC/VZPP

VZCC/VZPP skins are sets of parameters determining such VZCC/VZPP elements as the layout used (framed, non-framed, vzcc-plesk, vzcc-plesk, or plesk-integrated), the menu behavior (collapsing or static), as well as the icon and image files and CSS stylesheets used to form VZCC/VZPP pages. Multiple skins are available for VZCC/VZPP; you may also create and use your own skins. Any one of these skins may then be chosen on the **Configure Control Center** page of VZCC and the **Configure Power Panels** page of VZPP to be used in your browser.

In our example, we will define a new skin for VZCC. To this effect, you should perform the following operations:

- 1 In the Service VPS, open the `/etc/vzcp/skins.xml` file for editing:

```
# vi /etc/vzcp/skins.xml
```

- 2 Search for the following string in the file:

```
<mode id="vzcc">
```

- 3 Copy an existing `<skin>` element inside the file (all the text between the `<skin>` opening tag and the `</skin>` closing tag including these tags themselves) to provide a basis for your own skin. Your new `<skin>` element should be placed between the `<mode id="vzcc">` opening tag and the `</mode>` closing tag.

Note: While creating a new skin for VZPP, make sure your new `<skin>` element is placed between the `<mode id="vzcp">` opening tag and the `</mode>` closing tag inside the `/etc/vzcp/skins.xml` file.

- 4 Edit this `<skin>` element in the following way:
 - Between the `<id>` and `</id>` tags, enter a name that will be used to refer to your new VZCC skin. You may specify any name you consider suitable for your skin.
 - Between the `<theme>` and `</theme>` tags, provide a new string corresponding to the name of a subdirectory in the `/var/vzcp/static/vz/skins` directory housing all the necessary image and CSS files. There are a number of pre-installed subdirectories (aqua, ice, etc.) as is seen from the `<id>` elements of the existing skins. When using your own skin, you should make sure that you have created the corresponding subdirectory in the `/var/vzcp/static/vz/skins` directory. This subdirectory should, in its turn, contain the `icons` and `images` subdirectories with all the necessary icon and image files, correspondingly, and a custom CSS file named `vz.css` in its root. You should also make sure that all the icon and image files have names absolutely identical to those of any pre-installed skin and that your CSS file defines exactly the same styles as any pre-installed CSS file.
 - Between the `<name>` and `</name>` tags, provide a string that will be displayed when expanding the **Interface Skin** menu on the **Configure Control Center** page of VZCC.
 - Between the `<layout>` and `</layout>` tags, enter `framed`, `non-framed`, `vzcc`, `vzcc-plesk`, or `plesk-integrated`, depending on what kind of layout you wish to be associated with the current skin.
 - Between the `<menu>` and `</menu>` tags, enter either `collapsing` or `static`, depending on whether you want the left menu groups to be able to collapse or not for the current skin.
 - Between the `<buttons>` and `</buttons>` tags, enter either `images` or `text`, depending on whether you want the buttons on VZCC screens to be displayed as images found in the `/var/vzcp/static/vz/skins/<skin_name>/images` directory or to be drawn by the browser.
 - Optionally, add the `<background/>` element to enable beautiful backgrounds on VZCC pages.

- 5 Save the file.

Now you can choose the relevant skin for your browser by means of the **VZCC Configure Control Center** page.

Setting Up VZCC/VZPP Demo Mode for Potential Customers

The given section familiarizes you with the way to set up the demo mode for Virtuozzo Control Center and Virtuozzo Power Panels to allow your potential customers to try their functionality while managing real Virtual Private Servers and to customize these tools to enable these customers to access certain VZCC/VZPP features only.

Creating "Demo" User

There may happen situations when customers interested in buying one or more Virtual Private Servers wish to check the main VZCC/VZPP functionality and see how things look in practice before making the final decision on their purchase. To afford your potential customers such a possibility, you can resort to the help of the so-called VZCC/VZPP *demo mode*, i.e. configure Virtuozzo Control Center and Virtuozzo Power Panels in such a way as for your customers to be able to use all the VZCC/VZPP functionality and perform the same operations on Virtual Private Servers as a real VPS administrator except for the final steps causing any effectual changes inside VPSs. For example, users may be allowed to pass all the steps required for creating a Virtual Private Server such as choosing the VPS ID and OS template the VPS will be based on, customizing QoS parameters, and so on. However, after they click the **Create VPS** button on the last page of the **New VPS** wizard, the following message will be displayed: "Demo mode, operation not allowed".

Setting up the *demo mode* for VZPP and VZCC involves creating the so-called demo user that will possess all the aforementioned rights in respect of one or several Virtual Private Servers, respectively. Let us assume that you wish to set up the *demo mode* for VZCC and, therefore, are to create the Service VPS demo user. To this effect, you should perform the following operations:

- 1 Open the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS with the help of a text editor. For example, you can issue the following command on the Hardware Node to edit the file:

```
# vi /vz/root/1/etc/vzcp/vzcpcon.conf
```

- 2 In this file, locate the `<access>` element responsible for granting access to the VZCC/VZPP functionality and search for the following strings inside this element:

```
<auth>
  <type>SVEUser</type>
  <creds>
  ...
```

- 3 Before the `<auth>` opening tag, enter the text presented below to authorize the demo user in VZCC:

```
<!-- allow demo session for SVE -->
<auth>
  <type>SVEUser</type>
  <user>demo</user>
  <creds>
    <cred>
      <id>vzcp</id>
      <description>Top level cred</description>
      <policy>demo</policy>
    </cred>
  </creds>
</auth>
```

Note: 1. If you are creating the VZPP demo user, you should place the aforementioned text before the auth element containing the `<type>VEUser</type>` string.

2. If you have already created the Service VPS user for VZCC with the name of demo, you should specify another name for the VZPP demo user in the `/etc/vzcp/vzcpcon.conf` file (for example, `demo-pp`).

4 Save the file and restart vzcp for the changes to take effect:

```
# /etc/init.d/vzcp restart
Stopping vzcp: [ OK ]
Starting vzcp: [ OK ]
```

5 In Virtuozzo Management Console, create the demo Service VPS user. Please consult the [Creating Service VPS Users](#) subsection of the [Virtuozzo Installation Guide](#) to know how to create new Service VPS users in VZMC. While working with the demo Service VPS user, follow these guidelines:

- If you do not wish to allocate a separate Hardware Node to be managed by the Service VPS demo user, you may specify a range of VPSs (e.g. from 2000 to 3000) on an existing Node. In case you are going to dedicate a whole Node for the demo user, you may want this user to have access to all Virtual Private Servers residing on the Node (inclusive of the Service Virtual Private Server). To this effect, you should open the `/etc/vzagent.passwd` file inside the Service VPS with the help of any text editor (e.g. `vi`), add the `demo:1-99999` string to it, and save the file. In this case the demo user will be able to see and operate all Virtual Private Server in the range from 1 to 99999 including the Service VPS itself.
- Create several Virtual Private Servers in the specified range with different sets of applications, Quality of Service (QoS) resources, IP addresses, etc.
- Perform a number of operations on the created VPSs and make sure the corresponding logs are available on the **Status Changes** and **Actions Log** pages in VZCC. This will allow the demo user to view these logs by clicking the corresponding links on the VZCC menu and check how this functionality works in VZCC.

6 Log in to VZCC as the Service VPS demo user (i.e. using the credentials of the Service VPS user demo user that you have created in VZMC) and check that all critical VZCC functionality is running in the *demo mode*. First of all, you should pay attention to the following operations:

- rebooting the Hardware Node;
- creating a new Virtual Private Server;
- editing VPS network settings, etc.

After you have ascertained that all works fine, you can inform your potential customers of the IP address (or hostname) and TCP port to enter in the address line of their browser and of the demo user credentials (user name and password) to log in to VZCC.

Customizing VZCC/VZPP for Using in Demo Mode

The demo mode allows your potential customers to have access to all VZCC/VZPP functionality although your Hardware Node and Virtual Private Servers are ensured to be not affected while performing the critical operations on them (see the previous subsection for more information). However, you may customize Virtuozzo Control Center and Virtuozzo Power Panels to be used in the demo mode in one of the following ways:

- You can completely exclude those VZCC/VZPP features that might cause undesirable consequences on the Node and/or inside VPSs, i.e. you can prevent the corresponding menu items and buttons from displaying on the VZCC/VZPP menu and dashboards, respectively.
- You can enable the active mode for certain VZCC/VZPP features, i.e. allow your customers to perform some critical operations that cannot be performed (will complete without any result) in the default demo mode.

Let us assume that you wish to prevent the demo user from rebooting the Hardware Node, i.e. to make the **Reboot Hardware Node** button unavailable on the **Hardware Node** dashboard, and enable this user to start and stop their Virtual Private Servers, which is not permitted in the default demo mode. To this effect, you should do the following:

- 1 Open the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS with the help of a text editor. For example, you can issue the following command on the Hardware Node to edit the file:

```
# vi /vz/root/1/etc/vzcp/vzcpcon.conf
```

- 2 Search for the following strings in the file:

```
<auth>
  <type>SVEUser</type>
  <user>demo</user>
  <creds>
    <cred>
      <id>vzcp</id>
      <description>Top level cred</description>
      <policy>demo</policy>
    ...
```

The `<auth>` element in the code above is used to authorize the demo user in VZCC, and the `cred` elements are responsible for granting this user access to this or that VZCC functionality. For example, the top-level `cred` element containing the `<id>` subelement with the `vzcp` value provides access to the whole VZCC functionality. The `cred` elements are organized in the hierarchical form, i.e. a parent element includes a number of child elements, and child elements inherit the properties of their `<cred>` parent element. However, child elements may redefine the properties of their parent `<cred>` element (e.g. they can deny access to a certain VZCC feature).

Note: If the user has no right to access the parent `<cred>` element, s/he is not able to access any of its child `<cred>` elements.

- To prevent the **Reboot Hardware Node** button from being displayed on the **Hardware Node** dashboard in VZCC, you should add the `<cred>` element having `hw` as the value of the `<id>` subelement together with all its child `<cred>` elements after the `<policy>` element (see the code above). This `<cred>` element is responsible for granting access to all operations on the Node and looks as follows:

Note: A list of all `<cred>` elements available in the current version of Virtuozzo, is given in the `/etc/vzcp/credentials.xml` file inside the Service VPS. Please consult this file to find out and specify all `<cred>` elements defining those VZCC features that you wish the `demo` user to have access to.

```

...
<cred>
  <id>hw</id>
  <description>Hardware node operations</description>
  <cred>
    <id>management</id>
    <description>Hardware node management
operations</description>
    <cred>
      <id>license</id>
      <description>License management</description>
      <cred>
        <id>install</id>
        <description>Install new license</description>
      </cred>
      <cred>
        <id>request</id>
        <description>Generate request for
new license</description>
      </cred>
    </cred>
  </cred>
  <cred>
    <id>reboot</id>
    <description>Reboot hardware node</description>
  </cred>
  <cred>
    <id>config</id>
    <description>Change hardware node
config</description>
    <cred>
      <id>def-ostmpl</id>
      <description>Set default
OS Template</description>
    </cred>
    <cred>
      <id>def-vpstmpl</id>
      <description>Set default
VPS Template</description>
    </cred>
  </cred>
  <cred>
    <id>resources</id>
    <description>View hardware node resource</description>
  </cred>
</cred>
...

```

Now you should remove the `<cred>` element (i.e. all the code between the `<cred>` opening and `</cred>` closing tags including the tags themselves) containing the child `<id>` element having `reboot` as its value; this is the very `<cred>` element that is responsible for displaying the **Reboot Hardware Node** button on the **Hardware Node** dashboard.

- To allow the demo user to start and stop Virtual Private Servers, you should first add the `<cred>` element having `vps` as the value of the `<id>` subelement together with all its child `<cred>` elements. This `<cred>` element is responsible for granting access to all operations that can be performed in the VPS context and is given below:

```

...
<cred>
  <id>vps</id>
  <description>VPS operations</description>
  <cred>
    <id>management</id>
    <description>VPS management operations</description>
    <cred>
      <id>create</id>
      <description>Create new VPS</description>
    </cred>
    <cred>
      <id>destroy</id>
      <description>Destroy VPS</description>
    </cred>
    <cred>
      <id>start_stop</id>
      <description>Start, stop, fast stop and
        restart VPS</description>
    </cred>
    <cred>
      <id>enable_disable</id>
      <description>Enable and disable VPS</description>
    </cred>
    <cred>
      <id>reinstall</id>
      <description>Reinstall VPS</description>
    </cred>
    <cred>
      <id>repair_mode</id>
      <description>Start VPS in repair mode and return
        from repair mode</description>
    </cred>
    <cred>
      <id>autoinstall_plesk</id>
      <description>Autoinstall plesk to VPS</description>
    </cred>
    <cred>
      <id>change_root_password</id>
      <description>Change root password in
        vzpp mode</description>
    </cred>
    <cred>
      <id>properties</id>
      <description>Properties of VPS grouped by
        categories</description>
    </cred>
  </cred>
  <cred>
    <id>resources</id>
    <description>View resource</description>
  </cred>
  <cred>
    <id>backup</id>
    <description>List of backups</description>
  </cred>
  <cred>
    <id>application</id>
    <description>Allow install and delete application
      to/from VPS</description>
  </cred>
</cred>

```

```

        <id>services</id>
        <description>Services</description>
    </cred>
    <cred>
        <id>logs</id>
        <description>Logs, statuses and alerts</description>
    </cred>
</cred>
...

```

Now search for the `<cred>` element containing the child `<id>` element having `start_stop` as its value; this `<cred>` element is used to allow/forbid users to start and stop their Virtual Private Servers. Add the `<policy>` element before the `</cred>` closing tag and specify `allow` as its value (i.e. type `allow` between the `<policy>` opening and `</policy>` closing tags). So, your `<policy>` element should look as follows:

```

    <cred>
        <id>start_stop</id>
        <description>Start, stop, fast stop and
        restart VPS</description>
        <policy>allow</policy>
    </cred>

```

The `<policy>` element defines if the operations to which the `<policy>` element refers will be executed in the demo mode or in the active mode. In our case the `<policy>` element redefines the `<policy>` element with the demo value specified for the top level `<cred>` element (having `vzcp` as the value of its `<id>` child element) and enables the active mode for the start/stop operations.

3 Save the file and restart `vzcp` for the changes to take effect:

```
# /etc/init.d/vzcp restart
```

```
Stopping vzcp: [ OK ]
Starting vzcp: [ OK ]
```

Managing Demo Mode for VZPP With Plesk Integrated

You may wish to provide the VZPP demo user access to a Virtual Private Server with the Plesk control panel installed. In this case the demo user can log in to the Plesk control panel in the following ways:

- 1** By entering the VPS IP address and the TCP port of VZPP (by default, it is 4643), clicking the Plesk control panel link on the VZPP menu, and pressing the Login to Plesk button in the opened window.
- 2** By logging in to VZPP (as is shown in Point 1) and replacing the TCP port of VZPP with that of Plesk (by default, it is 8483) in the address line of your browser.
- 3** By using the VPS IP address and the TCP port of Plesk (by default, it is 8443).

To prevent the demo user from accessing the Plesk control panel, you should make sure that the `<cred>` element in the `/etc/vzcp/vzcpcon.conf` file inside the Service VPS responsible for logging in to Plesk is set to the demo mode. To this effect, you should search for the following section in this file:

```
<!-- demo session for VE -->
<auth>
  <type>VEUser</type>
  <user>demo-pp</user>
  <creds>
    <cred>
      <id>vzcp</id>
      <description>Top level cred</description>
      <policy>demo</policy>
    </cred>
  </creds>
</auth>
```

This section is used to authorize the VZPP demo user. You can check that the demo mode for Plesk is enabled in one of the following ways:

- If your section has only one `<cred>` element (as is shown above), you should ascertain that demo is specified as the value of the `<policy>` element inside this `<cred>` element.
- If there are several `<cred>` elements between the `<creds>` opening and `</creds>` closing tags, make sure that the `<policy>` element included in the `<cred>` element responsible for logging in to the Plesk control panel has the value of demo. In case this `<cred>` element includes no `<policy>` element in it, you should look for the nearest parent `<cred>` element containing the `<policy>` element and check that the demo value is specified between the `<policy>` opening and `</policy>` closing tags. In this case your section might be the following:

```
<!-- demo session for VE user -->
<auth>
  <type>SVEUser</type>
  <user>demo</user>
  <creds>
    <cred>
      <id>vzcp</id>
      <description>Top level cred</description>
      <policy>demo</policy>
    <cred>
      <id>vps</id>
      <decription>VPS operations</decription>
    <cred>
      <id>management</id>
      <decription>VPS management operations</decription>
    <cred>
      <id>start_stop</id>
      <policy>allow</policy>
    </cred>
  </cred>
  ...
  <cred>
    <id>services</id>
    <decription>Services</decription>
    <policy>allow</policy>
  <cred>
    <id>system_processes</id>
```

```

        </cred>
        <cred>
        <id>plesk_control_panel</id>
        <cred>
        <id>login</id>
        <policy>demo</policy>
        </cred>
        </cred>
    </cred>
    ...
</creds>
</auth>

```

As can be seen from the example above, the demo mode for the Plesk control panel is ensured by specifying the `<policy>demo</policy>` string inside the `<cred>` element responsible for logging in to Plesk (this `<cred>` element includes the child `<id>` element having `login` as its value). If the `<policy>` element is not specified for this `<cred>` element, the system looks for the `<cred>` pattern element, and now its `<policy>` element is taken into account and so on. In our case the next `<cred>` parent element containing the `<policy>` element is the following:

```

    ...
    <cred>
    <id>services</id>
    <decription>Services</decription>
    <policy>allow</policy>
    ...

```

The `<policy>` element has `allow` specified as its value, which means that the demo user will be allowed to log in to Plesk and act as a real Plesk administrator. Therefore, you should obligatory place the `<policy>` element inside the `<cred>` element used for enabling access to the Plesk control panel to prevent the demo user from accessing Plesk.

Important! While granting access for the demo user to the Virtual Private Server with the Plesk control panel installed, please always check that the demo mode for the Plesk control panel is enabled. Otherwise, the demo user will gain access to all Plesk functionality including the right to partially manage their Virtual Private Server (e.g. start, stop, and restart it) even though they are not allowed to do so in VZPP.

Importing Plesk Language Packs Into VZPP

Quite a number of localizations are built into VZPP by default. However, chances are that a new Plesk language pack is added to the Plesk installation in this or that VPS or an existing localization is updated. For VZPP to properly handle such a language pack, it should be imported into VZPP by the Hardware Node administrator.

Before importing, you should make sure that the information about the language in question is present in the VZPP languages database (which is considered to include almost all possible languages). Check the `/vz/root/1/etc/vzcp/languages.xml` file on the Hardware Node and see if the needed language is enumerated there. If it is not, you should create an additional `<language>` section in it for the new language. The structure and contents of this section should be similar to those present in the file.

To import a new or updated Plesk language pack, issue the following command on the Hardware Node:

```
# /vz/root/1/usr/libexec/vzcp/import-plesk-lp.sh -s VPS_ID -l lang_id
```

where *VPS_ID* is the ID of the VPS where the language pack is installed, and *lang_id* is the ID of the corresponding language (you can look it up in the `/vz/root/1/etc/vzcp/languages.xml` file).

After this shell script finishes its operation, the new Plesk localization becomes accessible by all the VPSs hosted on the Node.

CHAPTER 12

Advanced Tasks

In This Chapter

Configuring Capabilities	220
Migrating Physical Server to VPS	224
Migrating VPS to Physical Server	244
Changing System Time From VPS	248
Obtaining Hardware Node ID From Inside Virtual Private Server	249
Accessing Devices From Inside Virtual Private Server	250
Moving Network Adapter to Virtual Private Server	252
Enabling VPN for VPS	253
Loading iptables Modules	253
Sharing File System Among Virtual Private Servers	256
Creating Configuration File for New Linux Distribution	257
Rebooting Virtual Private Server	258
Managing Graphical Applications Inside VPS	259
Managing VZCC/VZPP Access Rights	265

Configuring Capabilities

Capabilities are sets of bits that permit of splitting the privileges typically held by the root user into a larger set of more specific privileges. The POSIX capabilities are defined by a draft IEEE standard (IEEE Std 1003.1e); they are not unique to Linux or Virtuozzo. When the Linux or Virtuozzo documentation says “requires root privileges”, in nearly all cases it really means “requires a specific capability”.

This section documents the tasks that can be achieved using per-Virtual-Private-Server capabilities in Virtuozzo and all configurable capabilities.

Creating VZFS Symlinks Inside VPS

Normally it is impossible to create a VZFS symlink from a Virtual Private Server. The ability to create VZFS symlinks presents a serious security concern explained further in this subsection. However, there may be a situation when you need such an ability, for example, for testing created templates or creating VZFS mounts.

A VZFS symlink is a symbolic link starting with four slashes. You can see VZFS symlinks in the private area of any Virtual Private Server, as is illustrated below:

```
# ls -l /vz/private/101/root/bin/bash
lrwxr-xr-x 1 root root 37 Jul 9 2001 \
/vz/private/101/root/bin/bash -> \
///redhat-7.2/bash-2.05-8/bin/bash
```

VZFS symlinks have no special meaning if the private area is not mounted over VZFS (to the VPS root directory). If it is, then instead of a VZFS symlink the users inside the VPS will see the file located in the template directory (in this particular case, /vz/template/redhat-7.2/bash-2.05-8/bin/bash) instead of the VZFS symlink.

If you try to create a VZFS symlink inside the Virtual Private Server, you will get an error:

```
[root@ve101 root]# ln -s ///redhat-7.2/bash-2.05-8/bin/bash .
ln: creating symbolic link './bash' to \
///redhat-7.2/bash-2.05-8/bin/bash': Invalid argument
```

The reason for this restriction is security considerations. If an intruder can correctly guess where the template area (defined by the `TEMPLATE` variable in the global Virtuozzo configuration file /etc/sysconfig/vz) is located, he/she can access any file on the Hardware Node provided the path to the file is guessed correctly. However, in case it is necessary to allow the VZFS symlinks creation inside a VPS, it is possible to make use of the `sys_rawio` capability:

```
# vzctl set 101 --capability sys_rawio:on --save
Unable to set capability on running VPS
Saved parameters for VPS 101
```

After restarting the Virtual Private Server, you can unpack VZRPMs inside the Virtual Private Server or simply create VZFS symlinks:

```
# ssh root@ve101
root@ve101's password:
Last login: Mon Oct 28 23:25:58 2002 from 10.100.40.18
[root@ve101 root]# rpm2cpio bash-2.05-8.i386.vz.rpm | cpio -id
94 blocks
[root@ve101 root]# ls -l bin/bash
-rwxr-xr-x 1 root root 519964 Oct 29 23:35 bin/bash
[root@ve101 root]# ln -s ///redhat-7.2/bash-2.05-8/bin/bash .
[root@ve101 root]# ls -l bash
-rwxrwxrwx 1 root root 519964 Oct 29 23:35 bash
```

As you can see both VZFS symlinks look like regular files for VPS users. If you need to unpack and work on symlinks themselves, you have to create a Virtual Private Server that has a directory bind-mounted over a regular file system such as EXT2FS, EXT3FS or ReiserFS.

Remember that assigning this capability to non-trusted Virtual Private Servers can lead to compromising the Hardware Node. The session below shows how a malicious VPS administrator can get a copy of the Hardware Node password database files:

```
[root@ve101 root]# ln -s ///.../etc/passwd .
```

```
[root@ve101 root]# ln -s ../../../../etc/shadow .
[root@ve101 root]# ls -l
total 3
-rwxrwxrwx 1 root root 1252 Oct 29 23:56 passwd
-rwxrwxrwx 1 root root 823 Oct 29 23:56 shadow
```

While there is no easy way to substitute the password files on the Hardware Node, a malicious VPS administrator could run a dictionary attack against the obtained files.

Available Capabilities for Virtual Private Server

This section lists all the capabilities that can be set with the `vzctl` command. The capabilities are divided into two tables: the capabilities defined by the POSIX draft standard and Linux-specific capabilities. For each capability, its description is given together with the default value for a Virtual Private Server.

Please note that it is easy to create a non-working Virtual Private Server or compromise your Hardware Node security by setting capabilities incorrectly. Do not change any capability for a VPS without a full understanding of what this capability can lead to.

Capabilities Defined by POSIX Draft

Name	Description	Default
<code>chown</code>	If a process has this capability set on, it can change ownership on the files not belonging to it or belonging to another user. You have to set this capability on to allow the Virtual Private Server root user to change ownership on files and directories inside the VPS.	on
<code>dac_override</code>	This capability allows to access files even if the permission is set to disable access. Normally leave this on to let the VPS root access files even if the permission does not allow it.	on
<code>dac_read_search</code>	Overrides restrictions on reading and searching for files and directories. The explanation is almost the same as above with the sole exclusion that this capability does not override executable restrictions.	on
<code>fowner</code>	Overrides restrictions on setting the <code>S_ISUID</code> and <code>S_ISGID</code> bits on a file requiring that the effective user ID and effective group ID of the process shall match the file owner ID.	on
<code>fsetid</code>	Used to decide between falling back on the old <code>suser()</code> or <code>fsuser()</code> .	on
<code>kill</code>	Allows sending signals to processes owned by other users.	on
<code>setgid</code>	Allows group ID manipulation and forged group IDs on socket credentials passing.	on
<code>setuid</code>	Allows user ID manipulation and forged user IDs on socket credentials passing.	on

Linux-Specific Capabilities

Name	Description	Default
setpcap	Transfer any capability in your permitted set to any process ID; remove any capability in your permitted set from any process ID.	off
linux_immutable	Allows the modification of the <code>S_IMMUTABLE</code> and <code>S_APPEND</code> file attributes. These attributes are implemented only for the EXT2FS and EXT3FS Linux file systems and, as such, this capability has no effect for Virtual Private Servers running on top of VZFS. However, if you bind mount a directory located on the EXT2FS or EXT3FS file system into a Virtual Private Server and revoke this capability, the root user inside the VPS will not be able to delete or truncate files with these attributes on.	on
net_bind_service	Allows to bind to sockets with numbers below 1024.	on
net_broadcast	Allows network broadcasting and multicast access.	on
net_admin	Allows the administration of IP firewalls and accounting.	off
net_raw	Allows to use the RAW and PACKET sockets.	on
ipc_lock	Allows to lock shared memory segments and <code>mlock/mlockall</code> calls.	on
ipc_owner	Overrides IPC ownership checks.	on
sys_module	Insert and remove kernel modules. Be very careful with setting this capability on for a Virtual Private Server; if a user has the permission of inserting kernel modules, this user has essentially full control over the Hardware Node.	off
sys_rawio	Allows to create VZFS symlinks over VZFS.	off
sys_chroot	Allows to use <code>chroot()</code> .	on
sys_ptrace	Allows to trace any process.	on
sys_pacct	Allows to configure process accounting.	on
sys_admin	In charge of many system administrator tasks such as swapping, administering APM BIOS, and so on. Shall be set to off for Virtual Private Servers.	off
sys_boot	This capability currently has no effect on a VPS behaviour.	on
sys_nice	Allows to raise priority and to set priority for other processes.	on
sys_resource	Override resource limits (do not confuse with user beancounters).	on
sys_time	Allows to change the system time.	off
sys_tty_config	Allows the configuration of TTY devices.	on
mknod	Allows the privileged aspects of <code>mknod()</code> .	on
lease	Allows to take leases of files.	on

Migrating Physical Server to VPS

This section provides information on how you can migrate an external physical server to a Virtual Private Server on your Hardware Node.

Migration Overview

Along with migrating Virtual Private Servers between your Hardware Nodes, you may wish to move a stand-alone physical server running a Linux operating system (Fedora Core, Debian, etc.) to a VPS on your Node. The migration process includes copying the whole contents of the physical server (i.e. all its files, directories, quota limits, configuration settings, and so on) to a VPS on the Hardware Node. After the server migration, you will have its exact copy in a VPS including the operating system running inside the VPS, the IP address(es) assigned to the VPS, the amount of available disk space and memory, etc.

You may wish to migrate a physical server to a VPS in the following situations:

- You wish to reduce administrative and operation costs related to your physical servers management and, therefore, decrease the number of physical servers under management by moving them into virtual ones. Multiple Virtual Private Servers on the Node can then be managed as efficiently as a single server; and many physical servers (Nodes) and all their VPSs can be managed centrally by means of VZMC.
- You wish to optimize resource utilization and move your physical servers, some of which might exploit their resources not to the full extent, to the Virtual Private Servers where each VPS is fully used and the resources are not wasted.
- You wish to minimize downtime costs. You move your physical servers to the Virtual Private Servers any of which can be easily and transparently backed up and rapidly deployed in the event of a system failure.

Migration Steps

Before you start migrating a physical server to a VPS on the Node, you should have a clear idea of the steps to be performed during the migration. The main steps of the migration procedure may be described as follows:

- 1 Creating the configuration file containing information on the main resources consumption on the physical server. This file is meant to be used for creating a Virtual Private Server on its basis. The data in the configuration file should be provided in the format readable by Virtuoizzo (i.e. in the form of "*PARAMETER*"="*value*"). Among other things, the file should include information on the Linux distribution your physical server is running and the number of user/group IDs allowed for VPS internal disk quota. Detailed information on quota limits and Linux distributions is provided in the **Managing Resources** chapter (on page 110) and in the **Linux Distribution Configuration Files** subsection of the **Virtuoizzo Reference Guide**, respectively.
- 2 Copying the configuration file made on the previous step from the physical server to the Hardware Node. You may copy the configuration file to any directory on the Node; the full path to this file should be specified during the physical server migration.

- 3 Creating a VPS on the basis of the configuration file copied to the Node. On this step, you are also supposed to specify an OS template to be used for creating the Virtual Private Server. Using an OS template for the VPS creation enables you to save RAM and disk space used by this VPS on the Hardware Node. In case an OS template is not specified, the `mkvzfs` command is executed during the VPS creation which makes an empty private area with the name of `/vz/private/VPS_ID` on the Node. On the next step, all the physical server files including its system and application files will be copied to the `/vz/private/VPS_ID` directory. Detailed information on OS templates is given in the [Understanding Templates](#) subsection (on page 21).
- 4 Migrating the physical server to the created VPS. During the server migration, the following operations are consecutively performed:
 - All the files, directories, etc. are copied from the server to the VPS on the Node by means of `rsync` - a utility providing the fast incremental data transfer. For more information on `rsync`, please see the man pages for this utility.
 - All the services on the physical server except for the critical ones (e.g. the `sshd` service needed to provide communication between the physical server and the Node) are stopped. This prevents the running services from modifying any files being moved. However, it depends entirely on you what services to stop.
 - The files, directories, etc. transferred to the VPS during the first `rsync` run are compared with those on the physical server and, if any changes to the files have been made during the files migration, they are copied to the VPS once more by means of `rsync` allowing to transfer just the differences between the two sets of files.

Note: If the migration process fails on this step, the `/vz/private/VPS_ID` directory on the Hardware Node will contain all the copied files and directories and may occupy a great amount of disk space. You can keep the directory, which will greatly speed up the repeated migration procedure, or manually remove the directory by using the `rm` utility.

- 5 Migrating the quota limits imposed on the selected partition from the physical server to the created VPS. You may specify only one partition on the physical server which will be migrated to the Virtual Private Server on the Node together with all quotas imposed on it. All the other partitions of the server will be copied without keeping their quota limits. Moreover, the quota limits of the migrated partition will be applied to the entire Virtual Private Server after the server migration. Detailed information on the quota limits is provided in the `vzquota` subsection of the [Virtuozzo Reference Guide](#) and in the [Managing Resources](#) chapter (on page 110).
- 6 Executing the post-migration scripts depending on the Linux distribution the physical server was running. The names of the scripts to be run are read from the corresponding distribution configuration file in the `/etc/sysconfig/vz-scripts/dists` directory on the Hardware Node. The scripts themselves are located in the `/etc/sysconfig/vz-scripts/dists/scripts` directory on the Node. They are needed to tune the VPS to be able to start it. Any script can be launched by executing the `vzctl runscript VPS_ID script_path` command on the Node where `VPS_ID` denotes the ID of the Virtual Private Server where the physical server has been migrated and `script_path` is the full path to the script on the Node.
- 7 Stopping the physical server and starting the VPS on the Node.

Virtuozzo allows you to complete all these steps in the following ways:

- 1 By using the `vzp2v` command line utility;

- 2 By using Virtuozzo Management Console (VZMC);
- 3 By using Virtuozzo Control Center (VZCC).

The aforementioned steps can be automatically performed while running the VZMC and VZCC migration wizards. However, if you wish to use the `vzp2v` utility to migrate a physical server to a VPS, you should manually create the configuration file by means of the `vzhwcalc` utility and copy it to the Hardware Node before starting the migration process itself. You may also use this utility previous to migrating a physical server in VZMC and/or VZCC to find out the resources consumption on the server during its maximal loading and set the right resources parameters on the corresponding steps of VZMC/VZCC wizards. Detailed information on the `vzhwcalc` utility and on how to create and modify the configuration file for the VPS where your physical server is to be migrated is provided in the **Preparing VPS Configuration File** subsection (on page 229).

Besides, while using `vzp2v`, you have to manually stop the physical server and start the VPS on the Node after the server migration whereas VZMC and VZCC allow you to select the corresponding options on the last step of their wizards.

The migration procedure by means of VZMC and the `vzp2v` utility is described in the following subsections; detailed information on how to migrate a physical server to a VPS by using VZCC is provided in the **VZCC Administrator's Guide**.

Migration Requirements

To avoid delays and problems while migrating your physical server to a Virtual Private Server on the Node, please make sure that the following requirements are fulfilled in respect of the server and the Hardware Node:

- The physical server is running a Linux distribution (Fedora Core, Red Hat, Debian, SUSE, Gentoo, etc.).

Note: None of the BSD operating systems is supported.

- The Linux distribution installed on the physical server is supported by Virtuozzo. To find out if your Linux distribution can be recognized by Virtuozzo, you can check the `/etc/sysconfig/vz-scripts/dists/` directory on the Node and look for the configuration file of your Linux distribution. It should have the name of `Linux_Distribution_Name-version.conf` where `Linux_Distribution_Name` and `version` denote the name of the Linux distribution running on your physical server and its version, respectively (e.g. `redhat-9.0.conf`). In case there is no corresponding distribution in the directory, you can proceed in one of the following ways:
 - Create a new distribution configuration file and place it to the `/etc/sysconfig/vz-scripts/dists/` directory on the Node. Detailed information on how to create new configuration files is provided in the **Creating Configuration File for New Linux Distribution** section (on page 257).

- Start the migration process without having the right configuration file for your Linux distribution. In this case the `unknown.conf` distribution configuration file from the `/etc/sysconfig/vz-scripts/dists/` directory on the Node will be used for tuning the VPS after the physical server migration. However, using the `unknown.conf` configuration file means that you will not be able to use standard Virtuozzo utilities (e.g. `vzctl`) for performing the main operations on the created VPS (such as setting the VPS IP address or configuring the DNS parameters) and have to manually complete these tasks from inside the VPS.
- The system libraries on the physical server are compatible with the 2.4 Linux kernel.
- A network connection can be established among your physical server, the Hardware Node, and the Service VPS on this Node.
- `ssh` is installed on both the physical server and the Hardware Node. `ssh` is used to provide secure encrypted and authenticated communication between the server and the Hardware Node. You can check if the `ssh` package is already installed on the server by executing the `ssh -V` command.
- `rsync` is installed on the physical server. `rsync` is used to copy the physical server contents to the Virtual Private Server. If the physical server `rsync` happens to be incompatible with the Hardware Node, use the statically linked `rsync` from the `usr/local/share/vzlinmigrate` directory on the physical server as well.
- The Service VPS exists on the Hardware Node and VZAgent is started. You can learn if VZAgent is started by executing the following command on the Hardware Node:
vzagent_ctl status
- The `vzhwcalc`, `vzlinmigrate`, and `vzlinmigrate-lib` packages are installed on the Hardware Node. During the Virtuozzo 3.0 installation or while upgrading your earlier Virtuozzo version to 3.0, these packages are automatically installed on the Node.

Migration Restrictions

Although Virtuozzo allows you to migrate virtually any physical server running a Linux distribution to a VPS, there is a number of limitations which should be taken into account before deciding on the migration process:

- During the migration, all the filesystems available on your physical server are joined to one filesystem inside the Virtual Private Server - VZFS (Virtuozzo File System). Detailed information on VZFS is provided in the [Virtuozzo File System](#) subsection (on page 18).
- If there are several IP addresses assigned to the physical server, all these IP addresses will be reassigned to one and the same device on the Node - `venet0` - a virtual network adapter used to connect all the VPSs on the given Hardware Node among themselves and with the Node.
- During the migration process, you may specify only one partition on the physical server which will be migrated to the Virtual Private Server on the Node together with all quotas imposed on it. All the other partitions of the server will be copied without keeping their quota limits. Moreover, the quota limits imposed on the selected partition will be applied to the entire VPS after the server migration.
- While migrating your physical server running a Linux operating system with the security-enhanced (SE) Linux kernel, please keep in mind that the SE Linux kernel is currently not supported by Virtuozzo. Therefore, you may get the VPS where the server with the SE Linux kernel has been migrated in a non-operational state.
- The current version of Virtuozzo does not support the Native POSIX Thread Library (NPTL) - a software feature that enables the Linux kernel to efficiently run programs written to use POSIX-style threads. So, although you can successfully migrate a physical server running a Linux kernel with the NPTL support to a VPS, you may have problems to start the applications using the NPTL-enabled libraries inside this VPS. For example, if your physical server is running Red Hat Enterprise 4 with NPTL and the MySQL 4 application with the Berkeley DB (BDB) tables 4 support is installed on the server, you might not be able to use this application inside the VPS after the physical server migration. This is caused by the fact that running MySQL 4 with BDB tables 4 requires NPTL to be supported by the Host OS on the Node.

You can still try to start NPTL-enabled applications inside the VPS by passing the `LD_ASSUME_KERNEL=2.4.19` parameter to the application environment. Generally, you should launch your program by this command:

```
# LD_ASSUME_KERNEL=2.4.19 your_program
```

There are a number of other ways to pass environment variables to various applications. Please consult the appropriate Unix guides for more information. Bear in mind that this method of enabling NPTL support in your VPS may still fail, if you had an incompatible version of the `glibc` library on your physical server.

- If any of your files and/or directories on the physical server have extended attributes associated with them, these attributes will be lost after the server migration.
- Raw devices on the physical server cannot and will not be migrated to the Virtual Private Server on the Hardware Node.
- If you are running an application which is bound to the physical server MAC address, you will not be able to run this application inside the VPS after the server migration. In this case, you can do one of the following:
 - If you are running a licensed application, you should obtain a new license and install the application inside the VPS anew.

- If you are running a non-licensed application, you can try to reconfigure the application and to make it work without being bound to any MAC address.
- If the migration process fails on the step of transferring files and directories from the physical server to the VPS by means of `rsync`, the `/vz/private/VPS_ID` directory on the Hardware Node will contain all the copied files and directories and may occupy a great amount of disk space. You can keep the directory, which will greatly speed up the repeated migration procedure, or manually remove the directory by using the `rm` utility.

Migrating Physical Server to VPS in Command Line

Preparing VPS Configuration File

If you wish to migrate a physical server to a VPS in the command line, i.e. by using the `vzp2v` utility, you should manually create the server configuration file and place it to the Hardware Node before starting the migration process itself. The configuration file contains information on the main server settings: its resource management parameters (e.g. disk space and the number of inodes consumed by the server, the server CPU power), network-related parameters (e.g. the server IP address and hostname), etc. During the physical server migration, information on the resources parameters from the configuration file is used to create a Virtual Private Server on their basis.

To prepare a configuration file for the physical server migration, you should perform the following operations:

- Copy the `vzhwcalc` utility from the Hardware Node to the server; you will need `vzhwcalc` to create the server configuration file.
- Copy the `distdetect-common.sh` script from the Hardware Node to the server; this script is used to determine the Linux version your server is running.
- Create the configuration file by running the `vzhwcalc` utility on the server.
- Edit the configuration file, if needed, and copy it to the Hardware Node.

As a result of the aforementioned operations, a valid configuration file should be created in the format readable by VirtuoZZo and copied to the Hardware Node. This file will be used to create a Virtual Private Server on its basis and the path to the file should be specified as the value of the `-c` option while running the `vzp2v` utility.

Creating VPS Configuration File

To create a configuration file of your physical server, you should first copy the `vzhwcalc` utility and the `distdetect-common.sh` script from the Hardware Node to the physical server. By default, `vzhwcalc` and `distdetect-common.sh` are stored in the `/usr/local/bin` and `/usr/local/share/vzlinmigrate` directories on the Node, respectively. The `vzhwcalc` utility is used to create a configuration file containing information on the server main resource parameters and used to create a VPS on its basis. In its turn, the `distdetect-common.sh` script is intended to determine what Linux distribution the server is running and to set the value of the `DISTRIBUTION` variable in the generated configuration file in accordance with the detected distribution. You may copy the `vzhwcalc` and `distdetect-common.sh` file to any directory on the physical server.

When launched, the `vzhwcalc` utility scans the main resources on your physical server, makes a snapshot of their consumption, and writes down this information to the server configuration file. Besides, the utility initiates the execution of the `distdetect-common.sh` script used to determine the Linux version installed on your server and to put this information to the generated configuration file.

So, after you have copied the `vzhwcalc` and `distdetect-common.sh` files to the physical server, you should run the `vzhwcalc` utility on it to create a configuration file for your server:

```
# vzhwcalc --scan-time time -p time -d script_path
```

where `--scan-time` is the time during which the `vzhwcalc` utility will be periodically making snapshots of the main server resources, `-p` denotes the interval with which the resources snapshots will be made by the `vzhwcalc` utility, and `-d` is the full path to the `distdetect-common.sh` script on the server. The time and interval should be given in the `dhms` format (e.g. `--scan-time 1d2h30m40s` means that the `vzhwcalc` utility will run on the server for 1 day, 2 hours, 30 minutes, and 40 seconds).

While running the `vzhwcalc` utility, please keep in mind the following:

- The consumption of the resources may significantly vary depending on the server loading. Therefore, we recommend that you set the scan time of the `vzhwcalc` utility to 1 day or more. During this time, the utility will periodically (i.e. with the interval specified) check the resources consumption on the server. As a result, the configuration file will be created on the basis of the peak values reached by the resources during the time specified. By default, all the resource parameters are calculated by `vzhwcalc` with a 150% allowance as compared to their maximal values (except for memory which is calculated with a 120% allowance compared to its maximal value). However, you can use the `--mem-scale` and `--disk-scale` options to set your own enlargement factor by which the calculated memory and disk space resources parameters will be increased in the configuration file.
- After executing `vzhwcalc`, you will be presented with a list of directories on the physical server which are highly recommended to be excluded from the migration process. The names of these directories should be given as the value of the `--exclude` option while running the `vzp2v` utility.
- During the `vzhwcalc` execution, the following warning messages may be displayed:

- A message informing you that the `distdetect-common.sh` script has failed to determine the Linux distribution your physical server is running. In this case you should manually specify your distribution name as the value of the `DISTRIBUTION` variable in the created configuration file. Detailed information on how to work with the `DISTRIBUTION` variable is provided in the next subsection.
- A message informing you that your physical server has two or more network interface cards installed. In this case all IP addresses assigned to several network interfaces on the server will be reassigned to one virtual network adapter on the Node - `venet0`. This virtual adapter will be used by the created VPS to communicate with the other Virtual Private Servers on the Node and with the outer world.
- A message containing a list of peer-to-peer IP addresses that cannot and will not be migrated to the Virtual Private Server to be created.
- A message informing you that the Linux OS installed on your physical server supports Native POSIX Thread Library (NPTL). For more information on NPTL, please see the **Migration Restrictions** subsection (on page 228).

The configuration file created by the `vzhwcalc` utility is placed to the same directory on the physical server from where you have run this utility and has the default name of `vps.conf`. However, you can pass the `-o` option to `vzhwcalc` and set a name of your choice for the resulting configuration file.

Editing VPS Configuration File

After you have created the VPS configuration file with the default name of `vps.conf`, you should check this file for the resources values listed in it. As has been mentioned above, the resource parameters in the configuration file are calculated on the basis of the physical server maximum load. However, you may wish to increase the resources available (e.g. in case you wish to exploit the VPS to be created more intensively than the physical server). You can do it by opening the `vps.conf` file for editing (for example, by means of `vi`) and entering new values for the corresponding parameters.

Along with editing the resource parameters, you should also look for the `DISTRIBUTION` variable in the configuration file used to define what post-migration scripts are to be executed depending on the Linux distribution set in this file:

- If the `DISTRIBUTION` variable is present in the file:
 - Make sure that the distribution configuration file whose name is indicated as the value of the `DISTRIBUTION` variable is present in the `/etc/sysconfig/vz-scripts/dists/` directory on the Node. All distribution configuration files have `.conf` as their extension added to the corresponding distribution name (e.g. `redhat.conf`).
 - In case there is no corresponding distribution configuration file in the `/etc/sysconfig/vz-scripts/dists/` directory, create a new distribution configuration file with the name specified as the value of the `DISTRIBUTION` value in the `vps.conf` file and place it to this directory. More information on the distribution file creation see below.

- If the `DISTRIBUTION` variable is absent in the file meaning that the Linux version running on the physical server could not be detected, you should do the following:
 - Create a new distribution configuration file for the Linux version running on the server and place it to the `/etc/sysconfig/vz-scripts/dists/` directory on the Node.
 - Specify the name of the newly created distribution configuration file as the value of the `DISTRIBUTION` variable in the `vps.conf` configuration file.

Detailed information on how to create new configuration files and set the `DISTRIBUTION` variable is provided in the **Creating Configuration File for New Linux Distribution** section (on page 257).

You can also start the migration process without having the right configuration file for your Linux distribution. In this case the `unknown.conf` distribution configuration file from the `/etc/sysconfig/vz-scripts/dists/` directory on the Node will be used for tuning the VPS after the physical server migration. However, using the `unknown.conf` configuration file means that you will not be able to use standard Virtuozzo utilities (e.g. `vzctl`) for performing the main operations on the created VPS (such as setting the VPS IP address or configuring the DNS parameters) and have to manually complete these tasks from inside the VPS.

Finally, you should copy the resulting configuration file to the Hardware Node. You will have to specify the full path to the configuration file while running the `vzp2v` utility.

Linux distribution installed on the physical server is supported by Virtuozzo. To find out if your Linux distribution can be recognized by Virtuozzo, you can check the `/etc/sysconfig/vz-scripts/dists/` directory on the Node and look for the configuration file of your Linux distribution. It should have the name of `Linux_Distribution_Name-version.conf` where `Linux_Distribution_Name` and `version` denote the name of the Linux distribution running on your physical server and its version, respectively (e.g. `redhat-9.0.conf`).

Migrating Physical Server to VPS

Now that you have created the configuration file and copied it to the Hardware Node, you can start the migration procedure itself. To migrate a physical server to a VPS, the `vzp2v` utility is used.

Let us assume that you wish to migrate a physical server running the Red Hat Enterprise Linux Advanced Server 3 (RHEL AS 3) operating system and having the IP address of 199.199.109.109 to VPS 101 on your Hardware Node; moreover, you are supposed to use the `root` user name and the `3e5rrt4` password to log in to the server. To this effect, you should issue the following command on the Node:

```
# vzp2v root@199.199.109.109 --vps-id 101 -c /etc/vps.conf \
-q /private_data -t -d redhat-as3 redhat-as3/20050217 \
--exclude=/proc/* --exclude=/usr/games -s iptables,cron
```

The options passed to the `vzp2v` utility in the example above are explained in the following table:

Option Name	Description
<code>--vps-id</code>	Mandatory. The ID of the VPS that will be created on the Node and where the physical server will be migrated. You can specify any unoccupied ID on the Node.
<code>-c</code>	Mandatory. The full path to the configuration file on the Node that was created on the physical sever by means of the <code>vzhwcalc</code> utility. You may specify only the name of the configuration file if you run the <code>vzp2v</code> utility from the directory where this file is located.
<code>-q, --quota</code>	Optional. The partition on your physical server which has any user and/or user groups quotas imposed on it. This partition will be migrated to the Virtual Private Server together with all quotas imposed on it. Moreover, these quotas will be applied to the entire VPS after the server migration.
<code>-d, --dist</code>	Optional. The Linux version your physical server is running. The name of the version specified should coincide with the name of the corresponding distribution configuration file located in the <code>/etc/sysconfig/vz-scripts/dist</code> directory on the Node. For example, if you specify <code>redhat-as3</code> as the value of this option, the <code>redhat-as3.conf</code> file should be present in the <code>/etc/sysconfig/vz-scripts/dist</code> directory on the Node. You should obligatorily set this option, if there is no <code>DISTRIBUTION</code> variable specified in the server configuration file. In case the <code>DISTRIBUTION</code> variable is set in the configuration file and you have specified the <code>-d</code> option, the latter takes precedence.

<code>-t, --ostmpl</code>	Optional. The OS template to be used to create the Virtual Private Server. You may list all OS templates installed on the Node together with their updates by executing the <code>vzpkgls</code> command. The names of OS templates usually correspond to those of Linux distributions (e.g. <code>redhat-as3/20050217</code> as in the example above where 20050217 denotes the OS template creation date); so you can easily guess what OS template to use for your Linux distribution. In case an OS template is not specified, the <code>mkvzfs</code> command is executed during the VPS creation which makes an empty private area with the name of <code>/vz/private/VPS_ID</code> on the Node. This private area is then used to copy all the physical server files to it.
<code>--exclude</code>	Optional. The path to the directories and files which will be excluded from copying to the VPS. This option allows you to avoid migrating the data you do not need. To gain more understanding on this option, please consult the man pages for the <code>rsync</code> utility.
<hr/> <p>Note: We strongly recommend that you exclude the files and directories you were informed of while running the <code>vzhwcalc</code> utility on the physical server.</p> <hr/>	
<code>-S, --srvstop</code>	Optional. The services to be stopped for the time of the physical server migration. We recommend that you stop all the services on the physical server except for the critical ones (e.g. the <code>sshd</code> service that is needed to provide communication between the physical server and the Node) before the migration. This will prevent the running services from modifying any files being moved.

In the example above, the following operations are performed during the physical server migration:

- 1 The `vzp2v` utility connects to the physical server with the IP address of 199.199.109.109 by using the `root` user name. While establishing a network connection, you will be asked for the password of `root` to log in to the server and have to enter `3e5rrt4` (which is, in our case, the password of the `root` user).
- 2 The `/etc/vps.conf` file is read and the `101.conf` file is created on its basis in the `/etc/sysconfig/vz-scripts` directory on the Node.
- 3 Virtual Private Server 101 is created on the basis of the `101.conf` file and the `redhat-as3/20050217` OS template.
- 4 All the data except for the `/usr/games` directory and the contents of the `/proc` directory is copied from the physical server to VPS 101.
- 5 The `iptables` and `crond` services are stopped on the physical server.
- 6 The files copied to VPS 101 are compared with those on the physical server and, if any changes to the files were made during the 4th migration step, these changes are copied to VPS 101.
- 7 The quota limits that were imposed on the `/private_data` partition on the physical server are copied to the VPS. These quota limits are applied to the entire VPS.
- 8 The post-migration script specific for the RHEL AS 3 OS is executed. The name of the script to be run is read from the `redhat-as3.conf` distribution configuration file located in the `/etc/sysconfig/vz-scripts/dists` directory on the Node and is needed to tune the VPS before its starting.

Migrating Physical Server to VPS in VZMC

VZMC provides a special wizard allowing you to quickly and reliably migrate a stand-alone physical server to a Virtual Private Server on your Node. You can launch the **Migrate Physical Server to VPS** wizard by right-clicking the **Virtual Private Servers** item under the **Hardware Node** where you wish to migrate the physical server and choosing **Migrate Physical Server to VPS** on the context menu. You will be presented with the following window:

Figure 48: VZMC - Logging In to Physical Server

The information you should enter in the fields provided is presented in the table below:

Field Name	Description
Server IP Address or Hostname	The IP address or hostname of the physical server you wish to migrate.
User Name	The user name used to log in to the physical server. You can specify the <code>root</code> user in this field, which is offered by default, or may use any other account to log in to the server. However, in the latter case you should make sure that the specified user has all the rights and privileges of the <code>root</code> user.

User Password

The password used to log in to the physical server by the user specified in the **User Name** field.

Clicking **Next** in the **Login to the Server being Migrated** window starts the process of connecting to the physical server and collecting information on the server configuration. The process is displayed in the progress bar of the **Collecting Server's Configuration** window. After the wizard has successfully connected to the physical server and finished collecting information on its configuration, the following window is displayed:

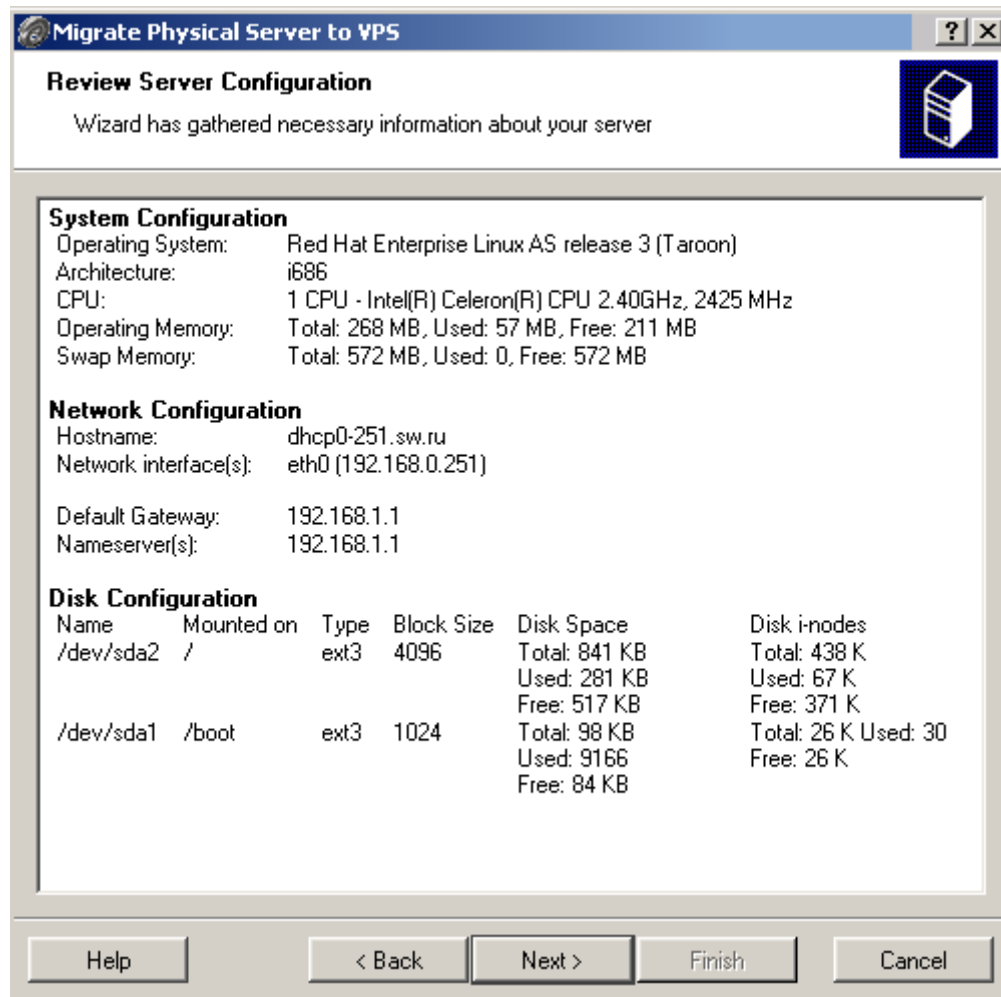


Figure 49: VZMC - Reviewing Server Configuration

The **Review Server Configuration** window allows you to check the configuration of the server you are going to migrate into a VPS. The information on the server is divided into three groups for your convenience:

- The **System Configuration** group including information on the operating system the server is running, the number and power of the processor(s) installed on the server, etc.
- The **Network Configuration** group containing information on the server hostname, the IP address(es) of the default gateway used by the server to access other networks, and so on.
- The **Disk Configuration** group holding data on the partitions that the physical server has: their name, type, disk space, etc.

After you have reviewed the information on the physical server configuration and clicked **Next**, the **Customize Server Migration** window is displayed:

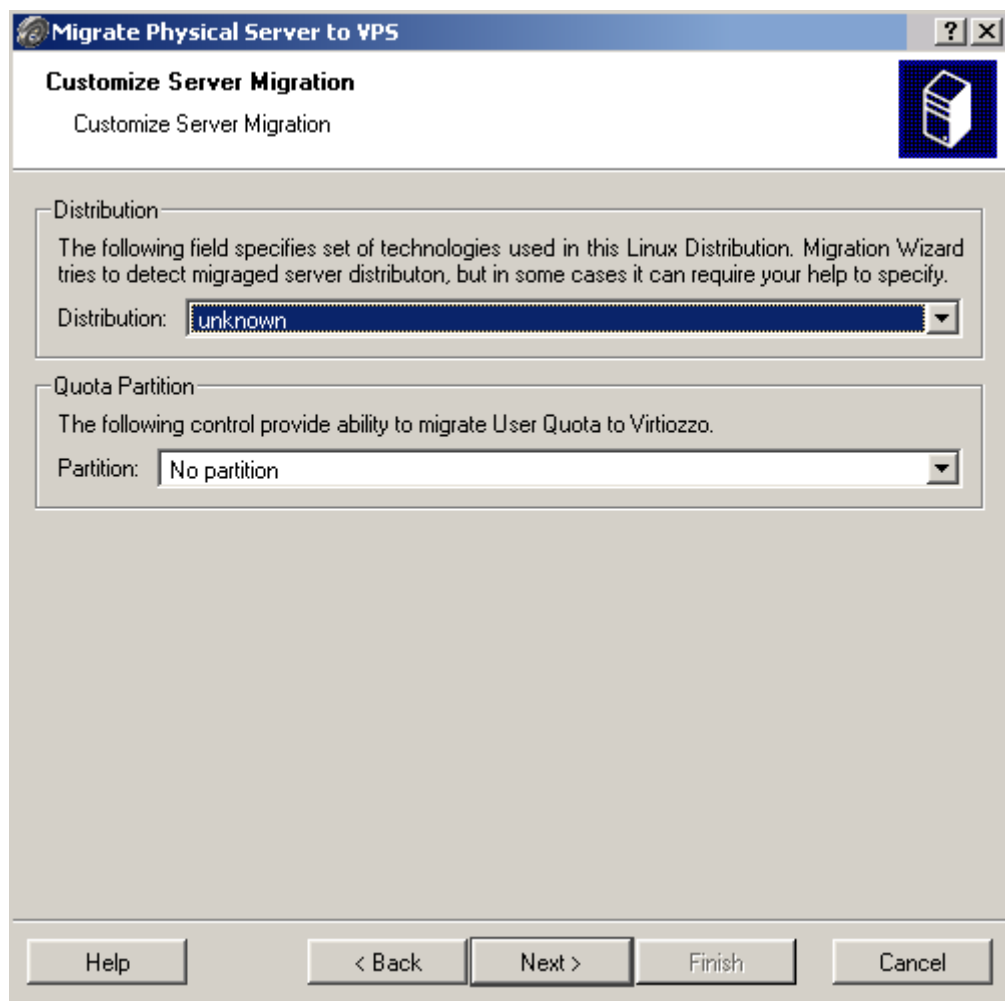


Figure 50: VZMC - Customizing Server Migration

In this window, you can perform the following operations:

- In the **Distribution** field, indicate the Linux distribution your physical server is running by selecting the right Linux version on the drop-down menu. The wizard tries to automatically determine the Linux distribution installed on your server and to offer the most suitable variant. If the wizard cannot specify what Linux distribution your server is running, the value of this field is set to "unknown". In this case you should manually select the corresponding Linux distribution on the drop-down menu; otherwise, you may get your VPS in a non-operational state after the physical server migration. In case you cannot find the right distribution on the drop-down menu, you can proceed in one of the following ways:
 - Select the most suitable distribution available on the Node. For example, if your physical server is running Fedora Core 3, you can choose `fedora-core-2` (the distribution configuration file for Fedora Core 2) or, if the latter is also lacking, `fedora-core` (the generic configuration file for all Fedora Core distributions). However, there is a slight chance that your VPS may not work properly due to some differences, which might be present in one and absent in another Linux version.

- Create a new distribution configuration file and place it to the `/etc/sysconfig/vz-scripts/dists/` directory on the Hardware Node. However, to be able to select this configuration file on the drop-down menu in the **Distribution** field, you should log off and log in to the physical server anew. You can do it either by closing the wizard and starting it again or by clicking on the **Back** button until you return to the **Login to the Server being Migrated** window and then proceeding with the wizard in the way described above. Detailed information on how you can create new distribution configuration files is provided in the **Creating Configuration File for New Linux Distribution** section (on page 257).
- In the **Partition** field, specify a partition on your physical server which has any user and/or user groups quotas imposed on it by selecting the right partition on the drop-down menu. The selected partition will be then migrated to the Virtual Private Server together with all quotas imposed on this partition. Moreover, the quota limits that were imposed on the selected partition on the physical server will be applied to the entire VPS after the server migration. For example, you might have created a number of user accounts having access to a certain partition on your physical server and set the maximal amount of disk space these users are allowed to consume within this partition. Specifying the name of this partition in the **Partition** field allows you to move the partition to the VPS and to keep all users disk space quotas imposed on it.

Note: 1. If your physical server has several partitions with quota parameters imposed on them, the quota parameters for all the partitions other than the one indicated in the **Partition** field will not be migrated. In this case you will need to manually set the corresponding quotas by means of VZMC or special Virtuozzo command line utilities after the physical server migration. Detailed information on how to manage the VPS quota parameters is provided in the **Managing Disk Quotas** section (on page 111).

2. Although the partition migration with quotas proceeds smoothly in most cases, we recommend that you check all the partition quotas after the physical server migration and adjust them, if needed.

When you are ready with specifying the right Linux distribution and partition, click **Next**.

The next screen allows you to exclude certain files and directories on the physical server from being migrated to the Virtual Private Server and, thus, to avoid copying the data you do not need. You may be already presented with a list of files and directories that are to be excluded from the migration process and that were automatically generated by the wizard. You can also use the **Add** and **Add Directory** buttons in the right part of the **Select Files and Folders to Exclude from Migration** window to additionally specify the files and directories you wish to exclude from being moved to the VPS. Click **Next**.

The **Stop Running Services Before Migration** window allows you to stop certain services for the time of the physical server migration:

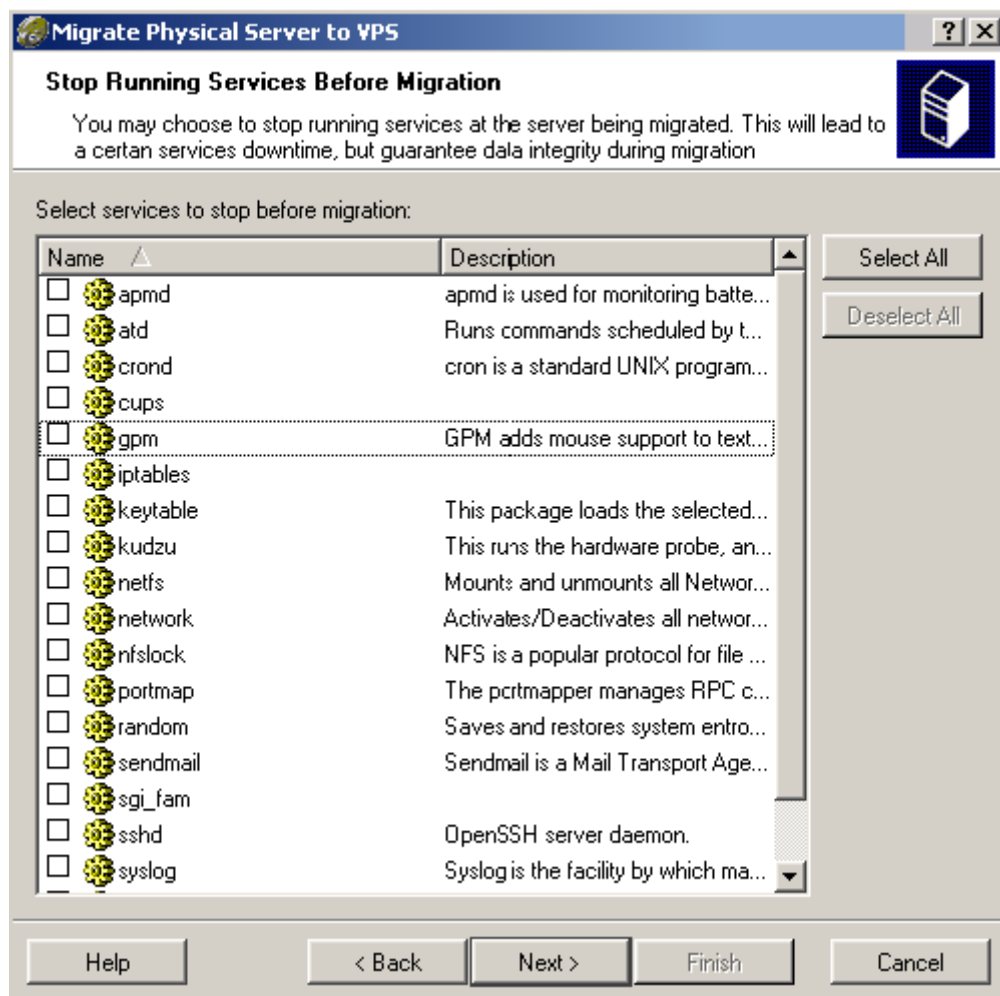


Figure 51: VZMC - Stopping Services

We recommend that you stop all the services on the physical server except for the critical ones (e.g. the `sshd` service that is needed to provide communication between the physical server and the Node) before the migration. This will prevent the running services from modifying any files being moved. All the services currently running on the physical server are listed in the table of the displayed window. To stop a service, select its checkbox. You can also make use of the **Select All** button in the right top corner of the window to stop all the services currently running on the physical server at once. After selecting the corresponding services for being stopped, click **Next**.

The next screen allows you to specify the VPS main parameters:

Migrate Physical Server to VPS

Specify Basic Virtual Private Server Parameters

The wizard needs to know basic parameters for destination Virtual Private Server

Virtual Private Server Configuration

☒ Use precalculated configuration

☐ Use following VPS sample configuration

Name	Description
vps.2048MB	Configuration file for allocating 2048 Mb of memory.
vps.256MB	Configuration file for allocating 256 Mb of memory.
vps.512MB	Configuration file for allocating 512 Mb of memory.
vps.basic	Basic configuration file
vps.basic1	
vps.confixx	Configuration file for running Confixx Control Panels
vps.cpanel	Configuration file for the VPS running cPanel Control Panel
vps.plesk7.rh9	redhat-9 with Plesk Server Administrator 7 application template

Virtual Private Server Settings

Virtual Private Server ID: 102

Hostname: dhcp0-39.sw.ru

Help < Back Next > Finish Cancel

Figure 52: VZMC - Specifying VPS ID and Hostname

In this window, you can do the following: should provide information in the following fields:

- Select the **Use precalculated configuration** checkbox to create the VPS by using the configuration file that was automatically generated by the wizard on the basis of the resources consumption on your physical server.
- Select the **Use following VPS sample configuration** checkbox to create the VPS on the basis of one of the VPS configuration sample files available on your Node. All the VPS sample files you can choose from are listed in the table in the centre of the displayed window. Detailed information on VPS configuration sample files is provided in the **Managing VPS Resources Configuration** section.
- **Virtual Private Server ID:** enter the ID of the Virtual Private Server which will be created on the Node and where the physical server will be migrated. Make sure that there is no VPS on the Node with the ID specified in this field.
- **Hostname:** enter the hostname of the Virtual Private Server which will be used to identify the VPS on a network.

After you have selected the corresponding checkbox and specified the VPS ID and hostname, click **Next**.

The **Specify OS Template** window allows you to choose an OS template and its version the Virtual Private Server will be based on. By default, VZMC automatically searches for the most compatible OS template. However, you can select any OS template listed in the table on this screen and create the VPS on its basis.

Clicking **Next** on the **Specify OS Template** screen displays the window where you are asked to specify the VPS network parameters:

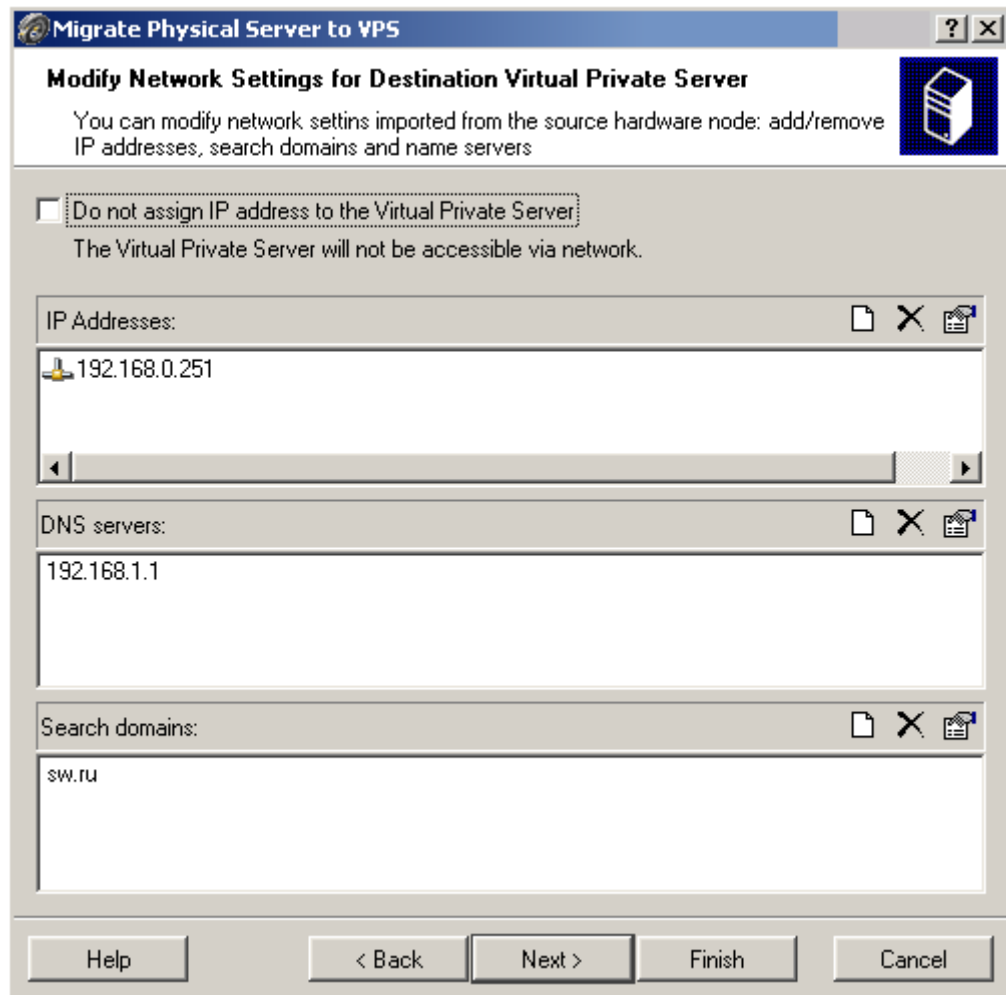


Figure 53: VZMC - Defining Network Parameters

In this window, you can do the following:

- Select the **Do not assign IP address to the Virtual Private Server** checkbox to create the VPS without assigning an IP address to it.
- If you clear the **Do not assign IP address to the Virtual Private Server** checkbox, you can use the provided **Add**, **Delete**, and **Edit** icons to perform the corresponding operations on the following VPS network parameters:
 - **IP Addresses**: use this field to manage IP addresses to be assigned to the Virtual Private Server. By default, the IP address(es) of the physical server is specified.
 - **DNS servers**: use this field to manage DNS servers that the VPS is supposed to use. By default, the DNS server(s) is shown which was used by the physical server to translate its qualified domain name into the IP address(es) and vice versa.

- **Search domains:** use this field to manage a list for hostname lookup. By default, the search domains of the physical server are shown. You can add other search domains for the VPS or delete any of the existing ones.

After you have set the VPS network parameters, click **Next** to open the window allowing you to adjust the QoS parameters for the VPS:

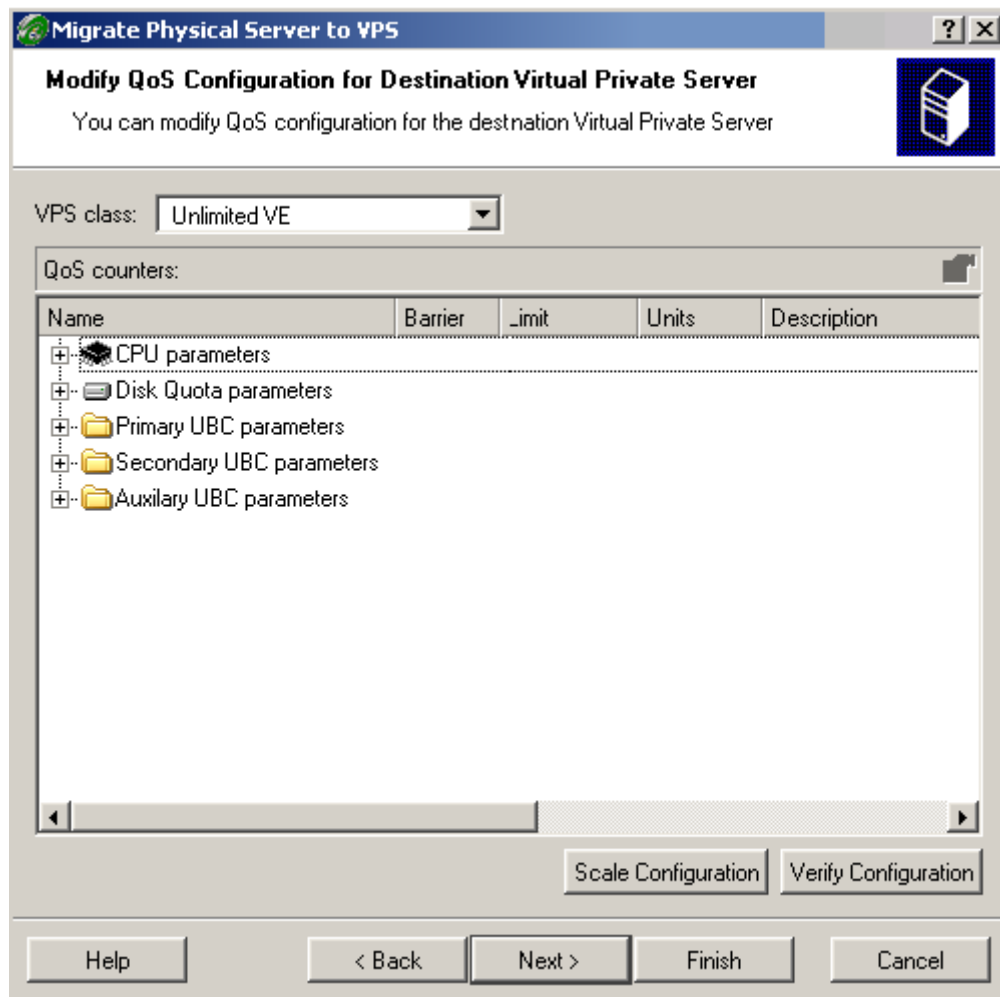


Figure 54: VZMC - Specifying QoS Parameters

All the resources are grouped by their relations to several subsystems for you to easier find information on the resource that interests you: **CPU parameters**, **Disk Quota parameters**, **Primary UBC parameters**, **Secondary UBC parameters**, and **Auxiliary UBC parameters**. Information on the VPS parameters is presented in the table having the following columns:

Column Name	Description
Name	The name of the QoS parameter.
Barrier	The quota on the consumption of the given resource by the VPS. In some situations, the system may allow the VPS to exceed this quota up to the limit.
Limit	The quota on the consumption of the given resource by the current VPS that cannot be exceeded in any circumstances.
Units	The units in which the value of the corresponding parameter is measured.

Description	The concise description of the given resource.
-------------	--

All the resource parameters shown in the table are calculated with a 150% allowance as compared to their original values (except for memory which is calculated with a 120% allowance to its original value), i.e. to those values that were collected by the wizard while scanning your physical server. However, you should keep in mind that the resources consumption on the physical server may significantly differ depending on its loading. So, you may need to increase the VPS resources parameters by double-clicking them and entering new values in the appropriate fields.

Note: While defining the right resources parameters, you can resort to the help of the `vzhwcalc` utility allowing you to scan the main resources on the physical server for a long period of time and to find out their consumption during its maximal loading. Detailed information on this utility is given in the **Creating VPS Configuration File** subsection (on page 229).

In the **Modify QoS Configuration...** window, you can also use the **Scale Configuration** and **Verify Configuration** buttons at the foot of the page to scale and verify the configuration of the Virtual Private Server, respectively. For information on how to scale and validate your existing configuration see the **Managing VPS Resources Configuration** section (on page 146). After you have made the necessary changes, click **Next**.

Detailed information on all Quality of Service parameters is provided in the **Managing Resources** chapter (on page 110). Please consult it to gather more understanding of the parameters.

The last screen of the wizard allows you to review the migration settings made on the previous steps. You can also compare the configuration of the physical server to be migrated with that of the Virtual Private Server to be created. Besides, you can select the **Shut down server and start VPS after migration** checkbox at the bottom of the screen to automatically stop the physical server and start the VPS after migration. This may be necessary to avoid the conflict of the physical and virtual servers due to the identical network settings. If you are satisfied with the parameters set, click **Finish** to start migrating the physical server to the VPS.

Note: If you press **Cancel** on certain steps, and the migration wizard exits, there may remain a temporary directory on the physical server that you should remove manually. The name of the directory is `/var/vzagent.tmp`.

Migrating VPS to Physical Server

You may also wish to migrate an existing Virtual Private Server on your Node to a physical server on a network. For example, this may be useful in case you migrated your physical server to a VPS on the Node, performed some operations on/inside this VPS (i.e. changed the content of some folders and directories), and now wish to move all the data (intact and changed) back from the VPS to the server.

Note: User quotas inside the Virtual Private Server are not migrated to the physical server in the current Virtuozzo version.

Migration Steps

The main steps performed while migrating a VPS to a physical server are the following:

- 1** A network connection is established between the Hardware Node and the physical server.
- 2** The Virtual Private Server to be migrated is set in the "stopped" and "mounted" state, in case it is running or stopped and unmounted.
- 3** A list of files and directories to be automatically excluded from the migration process is generated. The script used to create such a list depends on the Linux distribution the VPS is running. The name of the script is read from the distribution configuration file in the `/etc/sysconfig/vz-scripts/dists` directory and the script itself is located in the `/etc/sysconfig/vz-scripts/dists/scripts` directory on the Node. You can also specify additional files and directories that you do not wish to move to the physical server.
- 4** The files, directories, libraries, etc. are copied from the VPS to the physical server by using `rsync`. This utility allows you to transfer only the differences between the two sets of files, which significantly speeds up the process of copying data between the VPS and the server in case they differ only slightly from each other.
- 5** The `ldconfig` command is executed on the physical server. This command examines the copied shared libraries, if any, in the `/usr/local/lib`, `/usr/lib`, and `/lib` directories on the server and in the directories specified in the `/etc/ld.so.conf` file and updates the links and cache to these libraries. For more information on `ldconfig`, please see the man pages for this command.

The `vzv2p` utility used to migrate a VPS to a physical server allows you to automatically complete all the aforementioned tasks except for the last one, i.e. you have to manually run the `ldconfig` command on the server after the VPS migration.

Migration Requirements

Before starting the migration process, please make sure that your physical server and VPS meet the following requirements:

- A Linux distribution (Fedora Core, SUSE, Debian, etc.) is installed on your physical server. This distribution should correspond to that running inside the VPS you are going to migrate.
- A network connection can be established among your physical server, the Hardware Node, and the Service VPS on this Node.
- `ssh` is installed on both the physical server and the Hardware Node. `ssh` is used to provide secure encrypted and authenticated communication between the server and the Hardware Node. You can check if the `ssh` package is already installed on the server by executing the `ssh -V` command.
- `rsync` is installed on the physical server. `rsync` is used to copy the Virtual Private Server contents to the physical server. If the physical server `rsync` happens to be incompatible with the Hardware Node, use the statically linked `rsync` from the `usr/local/share/vzlinmigrate` directory on the physical server as well.
- The distribution configuration file for the Linux distribution running inside the VPS to be migrated is present in the `/etc/sysconfig/vz-scripts/dists` directory on the Node. The `DISTRIBUTION` variable in this file specifies what script is to be used to generate a list of files and directories, which will not be moved from the VPS to the physical server.

Migrating VPS to Physical Server

To migrate a VPS to your physical server, the `vzv2p` utility is used. Let us assume that you migrated your physical server to VPS 101 three months ago, the VPS was on the go during all this time (i.e. some of the old files and directories were changed, certain configuration settings modified, etc.), and now you wish to move VPS 101 back to your physical server. To this effect, you should issue the following command:

```
# vzv2p root@199.200.200.200 --vps-id 101 \
--exclude /home/private
```

The options passed to the `vzv2p` utility in the example above are explained in the following table:

Option Name	Description
<code>--vps-id</code>	The ID of the VPS on the Node to be migrated to the physical server.
<code>--exclude</code>	The directories to be excluded from being copied to the VPS. This option allows you to avoid migrating the data you do not need.

In our example, the `vzv2p` utility connects to the physical server with the IP address of `199.199.109.109` by using the `root` user name. While establishing a network connection, you will be asked for the password of `root` to log in to the server and have to enter `3e5rrt4` (which is, in our case, the password of the `root` user). After that, VPS 101 is brought to the "stopped" and "mounted" state and all the data except for the `/home/private` directory and directories that were automatically generated by the script defined on the basis of the `DISTRIBUTION` variable in the VPS configuration file is copied from VPS 101 to the physical server.

After the VPS has been successfully migrated to the physical server, you should execute the `ldconfig` command to update the links and cache to the shared libraries on the server.

Changing System Time From VPS

Normally it is impossible to change the system time from a Virtual Private Server. Otherwise, different Virtual Private Servers could interfere with each other and could even break applications depending on the system time accuracy.

Normally only the Hardware Node system administrator can change the system time. However, if you want to synchronize the time via Network Time Protocol (NTP), you have to run NTP software, which will connect to external NTP servers and update the system time. It is not advisable to run application software on the Hardware Node itself, since flaws in the software can lead to compromising all Virtual Private Servers on the Hardware Node. Thus, if you plan to use NTP, you shall create a special Virtual Private Server for it and configure it to have the `sys_time` capability. The example below illustrates configuring such a Virtual Private Server:

```
# vzctl set 101 --capability sys_time:on --save
Unable to set capability on running VPS
Saved parameters for VPS 101
```

The output of the above command warns you that `vzctl` cannot apply changes in the capabilities to a running Virtual Private Server. The VPS has to be restarted before changes take effect:

```
# vzctl stop 101; vzctl start 101
Stopping VPS ...
VE was stopped
VE is unmounted
Starting VPS ...
VE is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: ve101
VE start in progress...
# ssh root@ve101
root@ve101's password:
Last login: Mon Oct 28 23:25:58 2002 from 10.100.40.18
[root@ve101 root]# date
Mon Oct 28 23:31:57 EST 2002
[root@ve101 root]# date 10291300
Tue Oct 29 13:00:00 EST 2002
[root@ve101 root]# date
Tue Oct 29 13:00:02 EST 2002
[root@ve101 root]# logout
Connection to ve101 closed.
# date
Tue Oct 29 13:01:31 EST 2002
```

The command session above shows the way to change the system time from Virtual Private Server 101. The changes will affect all the Virtual Private Servers and the Hardware Node itself. It is not advisable to have more than one Virtual Private Server with the `sys_time` capability set on.

NTP is described in Internet Standard RFC 1305; more information including client software can be obtained from the NTP web server (<http://www.ntp.org>).

Obtaining Hardware Node ID From Inside Virtual Private Server

The default Virtuozzo installation does not allow users inside a Virtual Private Server to obtain any information specific to the Hardware Node the VPS is running on. The reason is that no Virtual Private Server shall have knowledge about the corresponding Hardware Node. A Virtual Private Server can be transparently migrated to another Hardware Node, and if this Virtual Private Server runs any applications depending on the particular Node, these applications might fail after the migration.

There are however situations when you have to provide some unique Hardware Node ID to some applications. For example, you might want to license your application per Hardware Node. In this case, after the migration your customer will need to re-apply the license for your application.

Virtuozzo provides access to the unique Hardware Node ID via the `/proc/vz/hwid` file. The default Virtuozzo installation makes this file accessible to Virtual Private Servers from 1 to 100 (i.e. VPSs with Virtuozzo-reserved IDs). It is possible to change this range in the Virtuozzo global configuration file. For example, this is the way to make the file visible in Virtual Private Servers from 1 to 1000:

```
# vi /etc/sysconfig/vz
VZPRIVRANGE="1 1000"
# vzctl exec 101 cat /proc/vz/hwid
0C3A.14CB.391B.6B69.02C9.4022.3E2F.CAF6
```

The above example illustrates accessing the Hardware Node ID from Virtual Private Server 101.

Accessing Devices From Inside Virtual Private Server

It is possible to grant a Virtual Private Server read, write, or read/write access to a character or block device. This might be necessary, for example, for Oracle database software if you want to employ its ability to work with raw disk partitions.

In most cases, providing access to the file system hierarchy for a Virtual Private Server is achieved by using bind mounts. However, bind mounts do not allow you to create new partitions, format them with a file system, or mount them inside a Virtual Private Server. If you intend to delegate disk management to a Virtual Private Server administrator, you shall use either the `--devices` or the `--devnodes` option of the `vzctl set` command.

The example session below illustrates the following situation: you want to allow the root user of Virtual Private Server 101 to take responsibility for administering the `/dev/sdb`, `/dev/sdb1` and `/dev/sdb2` devices. In other words, you allow the VPS 101 system administrator to repartition the `/dev/sdb` device and create file systems on the first two partitions (or use them with any software capable of working with raw block devices, such as Oracle database software).

First, we are going to grant the Virtual Private Server the permissions to work with the needed block devices:

```
# vzctl set 101 --devices b:8:16:rw --devices b:8:17:rw --devices
b:8:18:rw --save
Setting devperms
Saved parameters for VPS 101
```

This command sets the read/write permissions for block devices with major number 8 and minor numbers 16, 17 and 18 (corresponding to `/dev/sdb`, `/dev/sdb1`, and `/dev/sdb2`). If you are not sure which major and minor numbers correspond to the necessary block devices, you may issue the following command:

```
# ls -l /dev/sdb{,1,2}
brw-rw---- 1 root disk      8, 16 Jan 30 13:24 /dev/sdb
brw-rw---- 1 root disk      8, 17 Jan 30 13:24 /dev/sdb1
brw-rw---- 1 root disk      8, 18 Jan 30 13:24 /dev/sdb2
```

Now let us create a 100-Mb Linux partition in addition to an already existing 2 GB partition on `/dev/sdb1` from VE 101.

```
# [root@ve101 root]# fdisk /dev/sdb

Command (m for help): p

Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sdb1   *           1           255     2048256    83   Linux

Command (m for help): n
Command action
   e   extended
```

```

p   primary partition (1-4)
P
Partition number (1-4): 2
First cylinder (256-2231, default 256):
Using default value 256
Last cylinder or +size or +sizeM or +sizeK \
(256-2231, default 2231): +100M

Command (m for help): p

Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1    *           1          255     2048256    83  Linux
/dev/sdb2             256          268     104422+    83  Linux

Command (m for help): w

```

After the new partition table has been written, you can format it and mount inside the Virtual Private Server:

```

[root@ve101 root]# mke2fs /dev/sdb2
[Output of mke2fs is skipped...]
[root@ve101 root]# mount /dev/sdb2 /mnt
[root@ve101 root]# df

```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
vzfs	1048576	149916	898660	15%	/
ext2	101107	13	95873	1%	/mnt

Remember that you have to specify all minors for the devices you want to delegate authority for; allowing to access `/dev/sdb` grants the permission to create, modify and delete partitions on it, but explicit permissions shall be given for partitions you allow the Virtual Private Server to work with.

Moving Network Adapter to Virtual Private Server

By default, all the VPSs on a Node are connected among themselves and with the Node by means of a virtual network adapter called `venet0`. Starting with Virtuozzo 2.6.1, there is a possibility for a VPS to directly access a physical network adapter (for example, `eth1`). In this case the adapter becomes inaccessible to the Hardware Node itself. This is done with the help of the `vzctl` command:

```
# vzctl set 101 --netdev_add eth1 --save
Add network device: eth1
Saved parameters for VPS 101
```

Mind that the network device added to a VPS in such a way has the following limitations:

- This network device will be accessible only to the VPS whereto it has been moved, but not to the Hardware Node (VPS 0) and not to all the other VPSs on the Node.
- The port redirection mechanism is not supported for this network device.
- The Virtuozzo class-based traffic shaping, if set for the given VPS, does not limit the bandwidth for this network device.
- If such a device is removed from the VPS (by means of the `vzctl set --netdev_del` command) and added to another VPS instead, all the network settings of this device are purged. To work around this problem, you should store all the device settings in the `ifcfg-dev` file and have this file available in the `/etc/sysconfig/network-scripts` directory inside all the VPSs that may have access to this device (including VPS 0). After the device has been added to a VPS, it will be enough to issue the `ifup dev` command inside the VPS to read the settings from the file mentioned above. Mind though that this will still not restore advanced network configuration settings, such as traffic shaping or packet filtering rules.
- The physical device inside a VPS has no security restrictions typical for the `venet` virtual device. Inside the VPS it will be possible to assign any IP address to this device and use it, to sniff network traffic in the promiscuous mode, and so on.

Enabling VPN for VPS

Virtual Private Network (VPN) is a technology which allows you to establish a secure network connection even over an insecure public network. Setting up a VPN for a separate VPS is possible via the TUN/TAP device. To allow a particular VPS to use this device, the following steps are required:

- Make sure the `tun.o` module is already loaded before Virtuozzo is started:

```
# lsmod
```

- Allow the VPS to use the TUN/TAP device:

```
# vzctl set 101 --devices c:10:200:rw --save
```

- Create the corresponding device inside the VPS and set the proper permissions:

```
# vzctl exec 101 mkdir -p /dev/net
# vzctl exec 101 mknod /dev/net/tun c 10 200
# vzctl exec 101 chmod 600 /dev/net/tun
```

Configuring the VPN proper is carried out as a common Linux administration task, which is out of the scope of this guide. Some popular Linux software for setting up a VPN over the TUN/TAP driver includes Virtual TUNnel <<http://vtun.sourceforge.net/>> and OpenVPN <<http://openvpn.sourceforge.net/>>.

Loading iptables Modules

The current Virtuozzo kernel provides support for additional iptables modules that were not accessible in previous Virtuozzo releases. However, upgrading the Virtuozzo kernel does not have these new modules loaded automatically. If you want any of these modules to be loaded either to the Hardware Node or, additionally, to any particular VPSs, you should do some manual operations.

Loading iptables Modules to Hardware Node

To have certain iptables modules loaded on the Hardware Node startup, you should provide their names as the value of the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file. The default value of this parameter is the following:

```
IPTABLES_MODULES="ip_tables ipt_REJECT ipt_tos ipt_limit ipt_multiport
                  iptable_filter iptable_mangle ipt_TCPMSS ipt_tcpmss
                  ipt_ttl ipt_length"
```

You may modify this value to add any of the following modules:

```
ip_conntrack
ip_conntrack_ftp
ip_conntrack_irc
ipt_LOG
ipt_conntrack
ipt_helper
ipt_state
iptable_nat
ip_nat_ftp
ip_nat_irc
ipt_TOS
```

All the modules indicated as the value of this parameter will be loaded on the Node startup after you reboot the Hardware Node. However, if you want this set of modules to be loaded by default to the VPSs hosted on this Node or you wish to restrict loading any of these modules to all or particular VPSs, you should perform some additional steps.

Note: There are two modules that are always loaded to the Hardware Node: `ipt_REDIRECT` and `ipt_REDIRECT2`. Their loading inside VPSs is also preset: the `ipt_REDIRECT` module is never loaded to any VPS, and the `ipt_REDIRECT2` module is always loaded thereto.

Loading iptables Modules to Particular VPSs

What iptables modules are loaded by default inside the VPSs hosted on the given Node is determined by the value of the IPTABLES parameter in the `/etc/sysconfig/vz` file. Naturally, those modules that constitute the value of this parameter will be loaded to VPSs only in case they are also loaded on the Hardware Node itself (see page 254). This parameter can also be redefined both in VPS sample configuration files (`/etc/sysconfig/vz-scripts/ve-<sample_name>.conf-sample`) and in the configuration files of particular VPSs (`/etc/sysconfig/vz-scripts/<vps_id>.conf`).

In order to load extra iptables modules or not to load certain default modules inside particular VPSs, you should explicitly indicate what modules you *wish* to be loaded to these VPSs either by modifying the IPTABLES parameter in the respective VPS configuration files or by using the `vzctl` command. For example:

```
# vzctl set 101 --iptables iptable_filter --iptables ipt_length --
iptables ipt_limit --iptables iptable_mangle --iptables ipt_REJECT --
save
```

This command will tell Virtuozzo to load only the following modules to VPS 101: `iptable_filter`, `ipt_length`, `ipt_limit`, `iptable_mangle`, `ipt_REJECT`. This information will also be saved in the VPS configuration file thanks to the `--save` option.

Loading a new set of iptables modules does not happen on the fly. You should restart the VPS for the changes to take effect.

Notes: 1. There are two modules that are always loaded to the Hardware Node: `ipt_REDIRECT` and `ipt_REDIRECT2`. Their loading inside VPSs is also preset: the `ipt_REDIRECT` module is never loaded to any VPS, and the `ipt_REDIRECT2` module is always loaded thereto.

2. For the correct iptables modules operation inside a Virtual Private Server, these modules should be loaded inside the VPS before its start-up.

Sharing File System Among Virtual Private Servers

This section provides a simple example of what can be done with the help of Virtual Private Server action scripts. You need a basic BASH shell language knowledge to understand the examples.

Remember that when you source configuration files in your action script, you have two environment variables that show the path to VPS file areas: `$VE_ROOT` and `$VE_PRIVATE`. You need to use `$VE_ROOT` since the VZFS file system does not follow mount points in the VPS private area. In other words, if you mount a directory to the VPS private area, the users inside the VPS will not see this mount and you should use `$VE_ROOT` in your scripts.

This example shows how to create a configuration when two environments can share files and the necessary setup is automatically created at VPSs startup. Let us assume that both environments want to have their user home directories in sync. For the sake of simplicity, let VPS 102 (called test2) hold actual user directories and VPS 101 (called test1) use them as well.

In this case, VPS 102 does not need any action scripts. All the necessary setup is done by the mount script of VPS 101. It can look like the following:

```
#!/bin/bash
#
# 101.mount - script to mount home dir of VE 102

# if one of these files does not exist then something is
# really broken
[ -f /etc/sysconfig/vz ] || exit 1
[ -f "$VE_CONFFILE" ] || exit 1
[ -f /etc/sysconfig/vz-scripts/$veid.conf ] || exit 1

# If home dirs are not mounted we exit with error
mount --bind /vz/root/102/home $VE_ROOT/home
exit $?
```

This script is intentionally simplified to focus on the main idea of mounting one Virtual Private Server directories inside another. However, it can be developed further by adding checkups for the VPS 102 mount status (it is possible to call `vzctl` from the mount script, but do not call `vzctl` with the same VPS ID as the VPS the mount script is being executed for). It can source the VPS 102 configuration file to determine correctly the `VE_ROOT` directory of VPS 102.

In order to be able to stop VPS 101, you have to create the umount script dismounting `$VE_ROOT/home`:

```
#!/bin/bash
#
# 101.umount - a script to umount home directory of VE 102

# If one of these files does not exist then something is
# really broken
[ -f /etc/sysconfig/vz ] || exit 1
[ -f $VE_CONFFILE ] || exit 1
```

```
# Source configuration files to access $VE_ROOT
. /etc/sysconfig/vz
. $VE_CONFFILE

# Dismount shared directory
umount $VE_ROOT/home
```

After starting VPS 102 and 101, Virtual Private Servers will have a common /home directory.

It is possible to use the same technique for mounting the Hardware Node file system sub tree into a Virtual Private Server, to mount a block device into a Virtual Private Server (for example, a hard drive partition or a CD-ROM), and so on.

Creating Configuration File for New Linux Distribution

Distribution configuration files are used to distinguish among Virtual Private Servers running different Linux versions and to determine what scripts should be executed when performing the relevant VPS-related operations (e.g. assigning a new IP address to the VPS). Detailed information on distributions configurations files is provided in the [Linux Distribution Configuration Files](#) subsection of the [Virtuozzo Reference Guide](#).

All Linux distributions shipped with Virtuozzo have their own configuration files located in the `/etc/sysconfig/vz-scripts/dists/` directory on the Hardware Node. However, you may wish to create your own distribution configuration files to support new Linux versions released. Let us assume that you wish your Virtual Private Server(s) to run the CentOS-3 Linux version and, therefore, have to make the `centos-3.conf` distribution configuration file to define what scripts are to be executed while performing major tasks with VPSs running this Linux version. To this effect, you should do the following:

- 1 In the VPS configuration file (with the name of `/etc/sysconfig/vz-scripts/VPS_ID.conf`), specify `centos-3` as the value of the `DISTRIBUTION` variable (for example, `DISTRIBUTION="centos-3"`).
- 2 Create the `centos-3.conf` configuration file in the `/etc/sysconfig/vz-scripts/dists/` directory. The easiest way to do it is copy one of the existing configuration files by executing the following command in the `/etc/sysconfig/vz-scripts/dists/` directory:

```
# cp fedora.conf centos-3.conf
```

In the example above, we assume that the `fedora.conf` file is present in the `/etc/sysconfig/vz-scripts/dists/` directory on the Hardware Node. In case it is not, you may use any other distribution configuration file available on your Node.

- 3 Open the `centos.conf` file for editing with the help of any text editor:

```
# vi centos-3.conf
```

- 4 In the `centos-3.conf` file, go to the first entry and, in the right part of the entry, specify the name of the script you wish to be run on issuing the `vzctl` command with the parameter specified in the left part of the entry. For example, if you wish the script to be executed while assigning a new IP address to your VPS and the script has the `my_centos_script` name, your entry should look as follows:

```
ADD_IP=my_centos_script-add_ip.sh
```

Note: The information on all acceptable parameters and their description are provided in the Linux Distribution Configuration Files subsection of the Virtuozzo Reference Guide.

- 5 Repeat Step 4 for all entries in the file.
- 6 Place the scripts for the new Linux distribution to the `/etc/sysconfig/vz-scripts/dists/scripts` directory on the Node. Make sure the names of these scripts coincide with those specified in the `centos-3.conf` file.

Rebooting Virtual Private Server

When you issue the `reboot` command at your Linux box console, the command makes the reboot system call with argument `'restart'`, which is passed to the computer BIOS. The Linux kernel then reboots the computer. For obvious reasons this system call is blocked inside Virtual Private Servers: no Virtual Private Server can access BIOS directly; otherwise, a reboot inside a VPS would reboot the whole Hardware Node. That is why the `reboot` command inside a VPS actually works in a different way. On executing the `reboot` command inside a VPS, the VPS is stopped and then started by `vzagent`, which handles this situation.

If you want a Virtual Private Server to be unable to initiate reboot itself, add the `ALLOWREBOOT="no"` line to the Virtual Private Server configuration file (`/etc/sysconfig/vz-scripts/ve_id.conf`). If you want to have VPS reboot disabled by default and want to specify explicitly which Virtual Private Servers are allowed to reboot, add the `ALLOWREBOOT="no"` line to the Virtuozzo global configuration file (`/etc/sysconfig/vz`) and explicitly specify `ALLOWREBOOT="yes"` in the corresponding Virtual Private Server configuration files.

If the `vzagent` software is not running on your Hardware Node for this or that reason, an auxiliary way to allow VPSs to reboot themselves is to uncomment the following line in the `/etc/cron.d/vereboot` file:

```
# vi /etc/cron.d/vereboot
[beginning of file]
#* * * * * root /etc/sysconfig/vz-scripts/vereboot
```

You can use any editor of your choice instead of the `vi` command. Remove the hash mark on the last line to read:

```
* * * * * root /etc/sysconfig/vz-scripts/vereboot
```

Now you can issue the `reboot` command in a Virtual Private Server, and the latter will be started on the next `vereboot` run.

Managing Graphical Applications Inside VPS

The given section provides information on how you can run X applications inside your Virtual Private Servers located somewhere on a TCP/IP network and display them on your local computer, exploit window managers to customize the appearance of running X applications, and use the vnc desktop software to remotely launch graphical applications.

Running Graphical Applications in X Windows

Overview

You may wish to run X applications (X clients) such as `xclock`, `xmms`, etc. inside your Virtual Private Servers on a TCP/IP network and display the resulting output on your local computer. This can be done with the help of the X Window System. The X Window System is based on the client/server model where an X server is the program responsible for controlling the display of the computer on which you are working and an X client denotes an application program that communicates with the server, sending it various requests, such as "draw a line" or "pay attention to keyboard input".

To run X applications inside your Virtual Private Server located on a TCP/IP network and to display them on your local computer, you should take care of the following:

- Install and configure a special software called an X server on the computer where you wish X clients to be displayed.

Note: In the following subsections, we assume that you have successfully installed and configured an X server on your local computer. In case you have not, please download the X server software packages (e.g. from <http://www.xfree86.org>) and install them by following the instructions shipped with this software.

- Configure X clients (X applications) to direct their output to your local computer where the X server is running.
- You may also wish to specify a window manager of your choice to be used for displaying your X clients.

A central concept of the X Window System is the display, an abstraction for the screen managed by an X server. When an X client is invoked, it needs to know which display to use. Displays are named by strings in the form of *hostname:displaynumber.screennumber* and should be set as the `DISPLAY` environment variable on the computer where X clients are to be run (in our case inside the corresponding VPS):

- *hostname* specifies the hostname or the IP address of the machine to which the display is physically connected, i.e. the computer where the X server is running (e.g. `198.112.45.11:0.0`). An omitted hostname (e.g. `DISPLAY=:0.0`) would mean the local host.
- *displaynumber* is usually used to refer to a collection of monitors that share a common keyboard and pointer (mouse, tablet, etc.). Most workstations tend to have only one keyboard and pointer, and therefore, only one display. In case a workstation has several displays (i.e. several keyboards or pointer sets), each display on this computer is assigned a display number (beginning at 0) when the X server for that display is started. The display number must always be given in a display name.
- *screennumber*. Some displays share a single keyboard and pointer among two or more monitors. Since each monitor has its own set of windows, it is assigned a screen number (beginning at 0) when the X server for that display is started. If the screen number is not given, screen 0 will be used.

For example, if your local computer is known to the outside world as `my_local_computer` and located in the `my-domain.org` domain and you are running a normal X server on this computer, the value of the `DISPLAY` variable in the VPS environment where you wish to remotely run X clients should be set to `my_local_computer.my-domain.org:0.0`.

Using X Windows to Run Graphical Applications

The X Window System lets you start any X application inside any Virtual Private Server on a TCP/IP network and have it show up on your local computer where an X server is installed. To run remote X applications, you should first of all tell the X applications running inside your Virtual Private Server to direct their output to the display of your local computer. You can do it by specifying the `DISPLAY` environment variable inside the VPS. For example, to run the `xfig` drawing program inside your VPS and display its output on your local computer with the IP address of `199.199.199.199`, you should issue the following commands inside the VPS:

```
# DISPLAY=199.199.199.199:0
# export DISPLAY
# xfig &
```

Along with setting the `DISPLAY` environment variable inside your VPS, you should also open permissions to your X server so that X applications are allowed to use your local display. You can do it in one of the following ways:

- By using the host list mechanism (`xhost`). In this case the X server maintains a list of hosts which are allowed to connect to it.
- By using the magic cookie mechanism (`xauth`). In this case the X server allows access from any host having an authorization record (a magic cookie) stored inside the server.
- By forwarding X connections via `ssh`.

You can choose any of these ways to remotely run your X applications. However, by using the `xhost` and `xauth` mechanisms, authority records needed to establish a connection between an X server and X application are transmitted over the network with no encryption, whereas using `ssh` enables you to run X applications over encrypted connections. So, if you are worried that someone might snoop on your connections, you can use the X forwarding mechanism, as is shown in the example below.

Let us assume that you wish to run the `xclock` application inside VPS 101 and display its output on your local computer with the name of `my_local_computer.my-domain.org`. To this effect, you should perform the following operations:

Note: Before running X applications inside a VPS on a public network, check that this VPS is accessible from your local computer where the X server is to be run.

- 1 On your local computer, execute the `startx` command:

```
# /usr/X11R6/bin/startx
```

This starts an X server with a basic terminal window (the default `xterm` application) on your computer.

- 2 Once `xterm` is open, you should establish an `ssh` connection to a Virtual Private Server where you wish to run the `xclock` application:

```
# ssh VPS_IP_Address
```

where *VPS_IP_Address* denotes the IP address or hostname of the VPS where your X client is to be run. As has been mentioned above, an `ssh` connection is used to provide security and stronger authentication for an X protocol connection between the X server and the X client by tunneling the X protocol, which is called X forwarding. Moreover, X forwarding automatically sets the `DISPLAY` variable inside the VPS to point to your local computer and directs the output of X clients running inside the VPS to the X server on your local computer. X forwarding is enabled in `ssh1` and `ssh2` by default; however, you may additionally use the `-X` option to enable X forwarding in case you are not sure that it is on.

- 3 After executing the command, you will be prompted for the password to log in to the Virtual Private Server. Provide the `root` user name and their password to log in to the VPS and click Enter.
- 4 Now that you have successfully logged in to the VPS, execute the `echo $DISPLAY` command to check the value of the `DISPLAY` variable in your VPS environment. It should read: `my_remote_computer.swsoft.com:10.0`. As distinct from the `xhost` and `xauth` mechanisms where the display number in the `DISPLAY` variable reflects a real number of displays connected to a computer (beginning at 0), `ssh` always uses the 10th display number - a special X display created by `ssh` itself - to pass X protocol information to your local computer.

If you do not see any value when typing this command or the value is incorrect, set the `DISPLAY` variable in your VPS environment as follows:

```
# DISPLAY=my_remote_computer.swsoft.com:10.0
# export DISPLAY
```

- 5 Launch the `xclock` application displaying the current time in an analog form by issuing the following command:

```
# xclock
```

If a clock is shown on the screen of your remote computer, you have successfully run the `xclock` application.

Note: While running the commands in our example, we assume that you work in the `bash` shell. While working in other Linux shells, you may need to use different commands to start your X server or to set the `DISPLAY` variable on your local computer.

Defining Window Manager to Run X Applications

The layout of windows on the screen in the X Window system is controlled by special programs called window managers. Window managers (like `twm`, `wmaker`, `fvwm2`, etc.) are programs that sit between an X server and normal X clients and control the way the running X clients are positioned, resized, or moved on your screen. Although a window manager decides to a great extent how X clients look and feel, it does not affect what client applications do within the window defined by this window manager.

The main operations that can be performed by means of window managers are the following:

- Start and terminate X clients;
- Move, resize, and rearrange the "vertical" stacking of windows;
- Refresh the screen(s);
- Determine which window is to receive input from your keyboard or mouse;
- Create and customize pop-up menus to complete any of the aforementioned tasks, etc.

You can change the default window manager used to control the appearance of your X clients by editing the `Xclients` and `xinitrc` scripts located in the `/usr/X11R6/lib/X11/xinit/` directory either inside your VPS or on your local computer. However, you can launch only one window manager at any time. So, if you are already running a local window manager, you cannot start the remote one (i.e. it will complain and exit).

Let us assume that you wish to run several X applications (`xterm`, `oclock`, `emacs`) inside your VPS and to use the remote `fvwm2` window manager to manage their output on the screen. To this effect, you can edit the `/usr/X11R6/lib/X11/xinit/Xclients` script inside your VPS in the following way:

Note: We assume that you have successfully installed the `fvwm2` window manager inside your Virtual Private Server. In case you have not, please download the needed software packages (e.g. from <http://www.fvwm.org>) and install them by following the instructions shipped with this software.

- 1 Log in to your VPS and open the `/usr/X11R6/lib/X11/xinit/Xclients` file for editing:

```
vi /usr/X11R6/lib/X11/xinit/Xclients
```

This file is just a shell script containing commands that you wish to run when your X session starts (e.g. `xterm`, `xclock`).

- 2 Remove the existing text in the file and add the following strings to it:

Note: We recommend that you make a copy of the `Xclients` file in case something goes wrong.

```
#!/bin/sh
oclock -geometry 75x75-1-1 &
xterm -C -geometry 80x12+0+0 &
emacs &
fvwm2
```

The clients will be launched in the order in which they are listed in the file; the last line should specify the window manager where the started X clients will run.

- 3 Save the file.

In our example, the `Xclients` file starts three applications - `xterm`, `oclock`, and `emacs` - and the `fvwm2` window manager where these application are to be run. The `-geometry` options used in the example specify the size and shape of the window. `80x12+0+0` means a window that is 80 characters wide and 12 lines high, positioned at the upper left. The `+` and `-` numbers give the location of the window. The first number gives the X coordinate and the second one gives the Y coordinate. The `+` numbers start from the upper left of the screen; the `-` numbers start from the lower right of the screen. So, `+0+0` means to put the `xterm` application at the upper left corner. Numbers greater than 0 are used to put things in the middle of the screen as in case with the `oclock` window (a round clock) in our example.

Running Graphical Applications via VNC

You may also wish to use VNC (Virtual Network Computing) to remotely run graphical applications inside your Virtual Private Server and display them on your local computer. The main features of VNC are the following:

- The server and the client may be on different machines and on different types of computer. The protocol which connects the server and the viewer is simple, open, and platform independent.
- No state is stored at the viewer. Breaking the viewer's connection to the server and then reconnecting will not result in any loss of data. Because the connection can be remade from somewhere else, you have easy mobility.
- The VNC protocol is designed to adapt to the amount of bandwidth available which makes it ideal for thin client deployments.

To start using VNC, you should perform the following operations:

- Install a virtual X server - `vnc` - inside your VPS. The `vnc` servers are not associated with a physical display, but provide a "fake" one X clients (`xterm`, `mozilla`, etc.) can attach to.
- Install a `vnc` client - `vncviewer` - on your local computer to connect to the `vnc` server from anywhere on the network.
- Connect to the `vnc` server with the `vnc` viewer.

Let us run the `xclock` application inside VPS 101 with the hostname of `VPS101.com` located on a TCP/IP network and display it on your local computer by using VNC. To this effect, you should do the following:

Note: We assume that you have successfully installed a `vnc` server inside your VPS and a `vnc` client on your local computer. If you have not, please download the needed software packages (e.g. from <http://www.realvnc.com>) and install them by following the instructions shipped with this software or available on the web site.

1 Log in to VPS 101 and start your `vnc` server by issuing the following command:

```
# vncserver
```

If you have never run a `vnc` server before, you will be prompted for a password, which you will need to use when connecting to this server. All the `vnc` servers on your remote computer will use the same password; you can change it at a later time by using the `vncpasswd` command. Type the password you consider suitable and press `Enter`.

- 2 Execute the `echo $DISPLAY` command to check what display number will be used by the vnc server to run graphical applications. As you have learnt in the previous subsections, the main X display of a workstation is usually indicated as 0 (in our case it will read :0; the hostname is omitted because the vnc server is running inside the VPS itself). When you run a vnc server inside your VPS, it will appear as :1, as if it were just an additional display. Normally, the vnc servers will choose the first available display number and tell you what it is. However, you can specify your own display number (for example, 2) by typing the following:

```
# vncserver :2
```

You can also cause graphical applications to use a vnc server rather than the normal X display by setting the `DISPLAY` variable in the VPS environment to the vnc server you want (in the examples below, we assume that the display number for the vnc server is set to 2):

```
# export DISPLAY=VPS101:2
```

or by starting a graphical application with the `-display` option:

```
# xterm -display VPS101:2 &
```

- 3 Now you should connect the vnc viewer running on your local computer to the vnc server. You can do it by executing the following command on your local computer:

```
# vncviewer VPS101.com:2
```

where `VPS101.com` is the hostname of VPS 101 where the vnc server is running and 1 denotes the number of the display used by the vnc server to run graphical applications.

Note: While using hostnames for connecting to a Virtual Private Server, make sure that your VPS has a valid DNS entry. Otherwise, you should replace its hostname with the corresponding IP address.

You can control the way graphical applications are positioned, resized, or moved on the screen of your local computer by specifying different options for the `vncserver` command, as you do it by using window managers while running X applications. For example, you can pass the `-geometry` option to `vncserver` to set the size of the desktop to be created (by default, it is 1024x768). You can get a list of all options for the `vncserver` command by giving `-h` as its option.

Managing VZCC/VZPP Access Rights

Virtuozzo Control Center (VZCC) and Virtuozzo Power Panels (VZPP) can be used by many people including Hardware Node administrators, Service VPS users, and VPS administrators for managing the Node or any number of its VPSs in this or that way. It is possible to distinguish various VZCC/VZPP user groups and to allow them to perform different sets of actions by means of VZCC/VZPP. This is done by editing the `/etc/vzcp/vzcpcon.conf` configuration file located inside the Service VPS. This file is in the XML format, and all the possible access rights are defined by the `<access>` element together with its subelements.

Defining VZCC/VZPP User Groups

The `<access>` element in the `/etc/vzcp/vzcpcon.conf` file has a number of `<auth>` subelements, each of which defines the users constituting a group and the access rights this group has to VZCC/VZPP. Please follow these rules when defining user groups:

- Each `<auth>` element must have one and only one `<type>` subelement defining the type of the user group. The contents of the `<type>` subelement must be one of the following text strings: `SVEUser`, `VEUser`, `PleskAdmin`, or `RestorePassword`.
 - An `SVEUser` group comprises one or more Service VPS users. These users are logged in to Virtuozzo Control Center with appropriate permissions.
 - A `VEUser` group comprises one or more users of common VPSs. These users are logged in to Virtuozzo Power Panels of the corresponding VPS with appropriate permissions.
 - A `PleskAdmin` group comprises one or more administrators of the Plesk control panel installed inside a VPS. These users are logged in to Virtuozzo Power Panels with an integrated Plesk.
 - The `RestorePassword` group is an auxiliary group comprising one or more users for whom the functionality of restoring passwords is enabled. These users should also belong to one of the three groups mentioned above.
- There can be more than one `<auth>` element with the same `<type>` subelement.
- Each `<auth>` element can have one or more `<user>` subelements defining the names of the users belonging to this user group. If no `<user>` subelements are present, it means that all possible users of the given group type are considered, excluding those users that were indicated in other `<auth>` elements having the same group type. For example, one group of the `SVEUser` type has the `vzagent0` user defined, and if there were another group of this type with no users defined, it would include all Service VPS users but `vzagent0`. The same applies to the other group types.

With all user groups thus defined, you may assign distinct VZCC/VZPP permissions to each group.

Assigning Access Rights to VZCC/VZPP User Groups

Apart from the `<type>` and `<user>` subelements defining a user group, each `<auth>` element in the `/etc/vzcp/vzcpcon.conf` file must have one `<creds>` subelement, which defines specific VZCC/VZPP permissions of the given user group. The `<creds>` element consists of a number of subelements, each of which may consist of their own subelements, and so on. The last subelement of each branch is a single-tag element indicating that the corresponding permission is granted to the given user group. Thus, any user group has only those VZCC/VZPP access rights that are represented by single-tag elements in the `<creds>` element structure. This structure is described in the table below:

Element	Description
<code><creds></code>	Beginning of the section defining the user group permissions.
<code><hw></code>	List of Hardware Node-related permissions.

<code><management></code>	List of Hardware Node management permissions.
<code><license></code>	List of Hardware Node license permissions.
<code><install/></code>	Permission to install a new Virtuozzo license.
<code><request/></code>	Permission to request a new Virtuozzo license.
<code></license></code>	End of Hardware Node license permissions.
<code></management></code>	End of Hardware Node management permissions.
<code><resources/></code>	Permission to manage the Hardware Node resources.
<code></hw></code>	End of Hardware Node-related permissions.
<code><vps></code>	List of VPS-related permissions.
<code><management></code>	List of VPS management permissions.
<code><create></code>	List of VPS creation permissions.
<code><install_application/></code>	Permission to modify the list of applications to be installed into the VPS being created.
<code><qos/></code>	Permission to modify the QoS settings of the VPS being created.
<code></create></code>	End of VPS creation permissions.
<code><destroy/></code>	Permission to destroy the VPS.
<code><start_stop/></code>	Permission to start/stop/restart the VPS.
<code><reinstall/></code>	Permission to reinstall the VPS.
<code><repair_mode/></code>	Permission to repair the VPS.
<code><autoinstall_plesk/></code>	Permission to install the Plesk control panel, if it is not installed.
<code><change_root_password/></code>	Permission to change the VPS root password.
<code><properties></code>	List of VPS properties permissions.
<code><general/></code>	Permission to view and adjust the VPS general properties.
<code><network/></code>	Permission to view and adjust the VPS network properties.
<code><qos/></code>	Permission to view and adjust the VPS QoS properties.
<code></properties></code>	End of VPS properties permissions.
<code></management></code>	End of VPS management permissions.
<code><resources></code>	List of VPS resources permissions.
<code><advanced/></code>	Permission to view the current usage of VPS advanced resources.
<code><status_bar/></code>	Permission to view the resource status bar.
<code></resources></code>	End of VPS resources permissions.
<code><backup/></code>	Permission to back up and restore the VPS.
<code><application/></code>	Permission to install/remove applications to/from the VPS.
<code><services></code>	List of VPS services permissions.

<code><system_services/></code>	Permission to view and manage the VPS system services.
<code><system_processes/></code>	Permission to view and manage the VPS system processes.
<code><plesk_control_panel></code>	List of the Plesk control panel permissions.
<code><login/></code>	Permission to log in to the Plesk control panel.
<code><change_admin_password/></code>	Permission to change the Plesk administrative password.
<code></plesk_control_panel></code>	End of the Plesk control panel permissions.
<code><confiix_control_panel/></code>	Permission to use the Confixx control panels.
<code><ssh_connection/></code>	Permission to connect to the VPS by the ssh protocol (for Linux VPSs only).
<code><rdp_connection/></code>	Permission to connect to the VPS by the RDP protocol (for Windows VPSs only).
<code></services></code>	End of VPS services permissions.
<code><logs></code>	List of VPS logs permissions.
<code><status_changes/></code>	Permission to view the log of VPS status changes.
<code><qos_alerts/></code>	Permission to view the log of VPS QoS alerts.
<code><actions_log/></code>	Permission to view the VPS actions log.
<code></logs></code>	End of VPS logs permissions.
<code></vps></code>	End of VPS-related permissions.
<code><sample_vps_config></code>	List of permissions related to VPS sample configuration files.
<code><management></code>	List of VPS sample configuration files management permissions.
<code><create></code>	List of VPS sample configuration files creation permissions.
<code><application/></code>	Permission to adjust the list of installed applications in the VPS sample configuration file being created.
<code><qos/></code>	Permission to adjust the QoS parameters in the VPS sample configuration file being created.
<code></create></code>	End of VPS sample configuration files creation permissions.
<code><clone/></code>	Permission to clone VPS sample configuration files.
<code><delete/></code>	Permission to delete VPS sample configuration files.
<code><properties></code>	List of VPS sample configuration files properties permissions.
<code><general/></code>	Permission to adjust the general properties of VPS sample configuration files.
<code><qos/></code>	Permission to adjust the QoS parameters in VPS sample configuration files.
<code><application></code>	Permission to adjust the list of installed applications in VPS sample configuration files.
<code></properties></code>	End of VPS sample configuration files properties permissions.

<code></management></code>	End of VPS sample configuration files management permissions.
<code></sample_vps_config></code>	End of permissions related to VPS sample configuration files.
<code></creds></code>	End of the section defining the user group permissions.

All the subelements of any level are optional, i.e. you may leave out any particular permissions or groups of permissions from the list, in which case they will not be granted to the given user group. On the other hand, if you are planning to enable all the possible permissions within an element having both the opening and closing tags, you may replace a list of these permissions with the special `<all/>` subelement. For example, a list of permissions for the `vzagent0` user is as simple as that:

```
<creds>
  <all/>
</creds>
```

The `<all/>` subelement can be applied on any level of the `<creds>` hierarchy.

CHAPTER 13

Mastering VZMC

To leverage the full power of VZMC, it is important to be aware of those tasks that are much more convenient to perform thru the VZMC interface than thru the command line. The current chapter centers on the advanced VZMC features you can make use of while administering your Virtuozzo system.

In This Chapter

Managing Node Clusters	270
Managing Offline Services Configuration	276
Viewing Summary Pages	278
Managing Users and Groups	280
Configuring Firewall	282
Managing Mount Points	284
Viewing System and Virtuozzo Logs	285
Adjusting Template Configuration Files	286
Managing Files	288
Managing Search Domains	289
Managing Virtual Private Server Servers	289
Checking Virtual Private Server	290

Managing Node Clusters

Organizing Hardware Nodes into a cluster is used to ensure a unified space of VPS IDs and VPS IP addresses throughout all the Nodes in the cluster. A cluster has the following advantages in comparison with standalone Nodes:

- You do not have to bother anymore about the Virtual Private Server IDs when migrating VPSs between Nodes.
- You do not have to manually assign IDs and IP addresses when creating VPSs.
- You can easily check if there are any identical IDs or IP addresses assigned to Virtual Private Servers of different Nodes.

Each cluster shall have one Master Node, where the database of VPS IDs and IP addresses is stored, and any number of Client (Slave) Nodes constituting the cluster.

Note: A cluster can contain Hardware Nodes running both the Virtuozzo 32-bit version and the Virtuozzo 64-bit version for x86-64 processors.

Creating Cluster

To create a cluster of Nodes, you should right-click the Node you wish to be the Master one and select **Tasks --> Create Virtuozzo Cluster** on the context menu.

The newly-created Virtuozzo cluster will be displayed in the left pane of the VZMC window on the same level as standalone Nodes. The name of the cluster will be put in parentheses and coincide with the name of its Master Node. Click on the cluster name to view a list of Nodes registered in it together with vital statistics on them in the right pane. The information will be presented as follows:

Field Name	Description
Name	The name of the Hardware Node.
Address	The IP address of the Service VPS.
Connected with Master	Indicates whether the corresponding Node is connected to the Master Node, i.e. it can exchange data with the Master Node. The Node may happen to be the Master Node itself.
License	The status of the Virtuozzo license.
Agent	The version of the vzagent software.
OS	The operating system installed on the Hardware Node.
Memory	The amount of the physical memory installed on the Node.
Free Memory	The current amount of free physical memory on the Node.
Connected	Whether the given Node is connected to VZMC.

You may want to destroy a cluster. To this effect, select **Tasks --> Destroy Virtuozzo Cluster** on the context menu.

Note: After you destroy a cluster, all configured resource pools containing in the cluster will also be deleted.

Adding Nodes to Cluster

The Nodes that are added to an existing cluster automatically become Client (Slave) Nodes, as any cluster can have only one Master Node for storing a centralized database of VPS IDs and IP addresses. The easiest way to add new Nodes to a cluster is the following:

- 1 Right-click the name of the cluster where you wish to add new Nodes and select **Add Hardware Node to Cluster** on the context menu.
- 2 In the **Add Client Hardware Node Into Cluster** window, select those Nodes that you wish to add. In case of a large number of Nodes, it is reasonable to use the **Select All** and **Deselect All** buttons. The **Register New** button lets you invoke the **Register Hardware Node** wizard right from this window and then add the newly-registered Node to the cluster.
- 3 Press the **OK** button.

If you wish to add a particular Node to an existing cluster, you may also perform the following steps:

- 1 Right-click the name of the Node that you wish to add to a cluster and select **Tasks --> Register Node in Cluster** on the context menu.
- 2 In the **Add Client Hardware Node Into Cluster** window, select a Node in the table having "Master node" as its cluster status. If no Master Nodes are shown, it means that there are no clusters registered in VZMC. In this case you should close the **Add Client Hardware Node Into Cluster** dialog window and create a cluster first.
- 3 Press the **Register** button.

Configuring IP Addresses Pool

After you have registered all the needed Hardware Nodes in a cluster, it is time to configure the IP Addresses and IDs pool for Virtual Private Servers hosted on the Nodes of the given cluster. One of the purposes of organizing Hardware Nodes into clusters is to ensure a unified space of VPS IP addresses throughout all the Nodes in the cluster. The database of the IP addresses to be assigned to the VPSs belonging to all the Nodes of the cluster is stored on the Master Node.

To configure the pool of IP addresses, do the following:

- 1 Right-click the corresponding cluster name and select **Configure IP Addresses Pool** on the context menu.
- 2 On the **Pool Configuration** tab of the **IP Addresses Pool Configuration** window, view and configure one or more of continuous ranges of IP addresses comprising the pool. Every range may be characterized by the starting IP address, the ending IP address, and the number of IP addresses within the range. Obviously, it is enough to know any two of these three parameters to deduce the third one. This information on all the defined ranges is presented in the table. To configure the pool as desired, use the buttons to the right of the table:

Button	Description
Add Range	Displays a window where you can define a new range for the IP addresses pool of the current cluster.
Delete	Deletes the IP addresses range selected in the table.
Edit	Displays a window where you can edit the parameters of the range selected in the table.
Exclude Range	Displays a window where you can exclude a certain continuous subrange of IP addresses from the range selected in the table. As a rule, this brings about the appearance of two new ranges instead of the selected one.

Configuring Virtual Private Server IDs Pool

Another purpose of organizing Hardware Nodes into clusters is to ensure a unified space of VPS IDs throughout all the Nodes in the cluster. This would allow to avoid, for example, migration problems. The database of the IDs to be assigned to the VPSs belonging to all the Nodes of the cluster is stored on the Master Node.

To configure the pool of Virtual Private Server IDs, do the following:

- 1 Right-click the corresponding cluster name and select **Configure VPSIDs Pool** on the context menu.
- 2 On the **Pool Configuration** tab of the **Virtual Private Server IDs Pool Configuration** at Node window, view and configure one or more of continuous ranges of IDs comprising the pool. Every range may be characterized by the starting ID, the ending ID, and the number of IDs within the range. Obviously, it is enough to know any two of these three parameters to deduce the third one. This information on all the defined ranges is presented in the table. To configure the pool as desired, use the buttons to the right of the table:

Button	Description
Add Range	Displays a window where you can define a new range for the VPS ID pool of the current cluster.
Delete	Deletes the ID range selected in the table.
Edit	Displays a window where you can edit the parameters of the range selected in the table.
Exclude Range	Displays a window where you can exclude a certain continuous subrange of VPS IDs from the range selected in the table. As a rule, this brings about the appearance of two new ranges instead of the selected one.

Specifying Cluster Network Settings

VZMC allows you to define DNS servers for all Virtual Private Servers residing on Hardware Nodes that constitute a cluster. It can be of great use, for example, while creating a new Virtual Private Server. In this case the DNS servers field will be automatically filled in with the corresponding values from your cluster.

To set the aforementioned parameters, perform the following operations:

- 1 Right-click the cluster name and select **Network Configuration** on the context menu. The **Virtuozzo Cluster Network Setting** window appears:

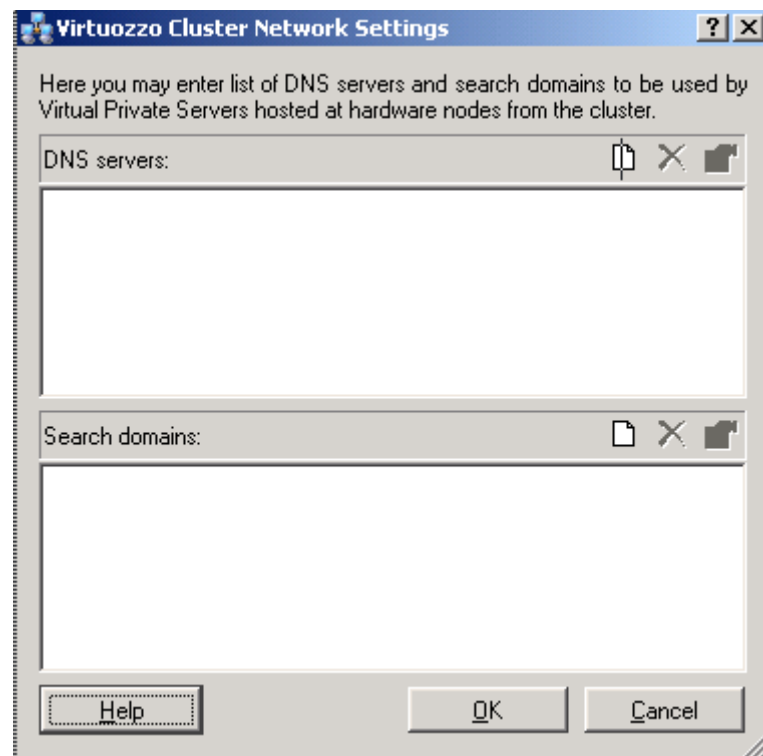


Figure 55: VZMC - Setting DNS Servers

- 2 Use the **Add**, **Remove**, and **Properties** buttons to the right of the **DNS servers** and **Search domains** fields to set, remove a DNS server or search domain, or change their properties, respectively.
- 3 After you have made the necessary changes, click **OK**.

Checking for Conflicts Among Nodes

There may happen situations when the consistency of really assigned VPS IDs and IP addresses across all the Nodes in the cluster is broken, i.e. when they do not correspond to the database configuration stored on the Master Node.

The most common kind of such conflicts happens when a newly-created VPS receives an ID or an IP address that have not been included in the corresponding pool. These conflicts are not very dangerous, but they may create problems in the future, so it is better to resolve them once they have been identified.

More serious conflicts are represented by one and the same IDs or IP addresses assigned to separate Virtual Private Servers. This situation may lead to immediate problems with VPSs functioning, so you should get rid of identical IDs or IP addresses at once.

To check for conflicts in the given cluster, right-click the cluster name in the left pane of the VZMC window and select **Check for Conflicts** on the pop-up menu. The **Virtual Private Server Conflicts Resolution** window is displayed.

This window identifies the conflicts described above and allows you to solve them. Depending on whether there are VPS ID conflicts or IP addresses conflicts, or both, the corresponding tabs are available in the window.

Whatever the tab, a table in the upper part of the window lists all the found conflicts on separate lines. The yellow icon denotes that the conflict is not that serious, and the red icon - that the conflict calls for an immediate resolution.

In this window, you can perform the following tasks:

- To sort all the conflicts in the table by a certain column, click on the heading of the column. A second click on the same heading reverses the sorting order.
- To view the detailed description of a conflict and its default resolution, select the corresponding line in the table. The description will be provided in a text area under the table.
- To resolve the desired conflicts, select the corresponding checkboxes in the table and press the **Resolve** button. To facilitate the selection, you may use the three buttons to the right of the table:
 - Pressing the **Select All** button marks all the checkboxes;
 - Pressing the **Deselect All** button clears all the checkboxes;
 - Pressing the **Default Resolution** button marks those checkboxes that allow for the default resolution of the corresponding conflicts.

Managing Offline Services Configuration

VZMC allows you to configure the properties of the existing offline services as well as add new services and delete those services that are not needed any more. This is done in the **Offline Services Configuration** window that can be displayed by right-clicking the corresponding Hardware Node name and selecting **Tasks -> Manage Offline Management Configuration** on the context menu:

Note: Detailed information on offline services is provided in the **Setting VZCC/VZPP to Work** chapter of the **Virtuozzo Installation Guide**.

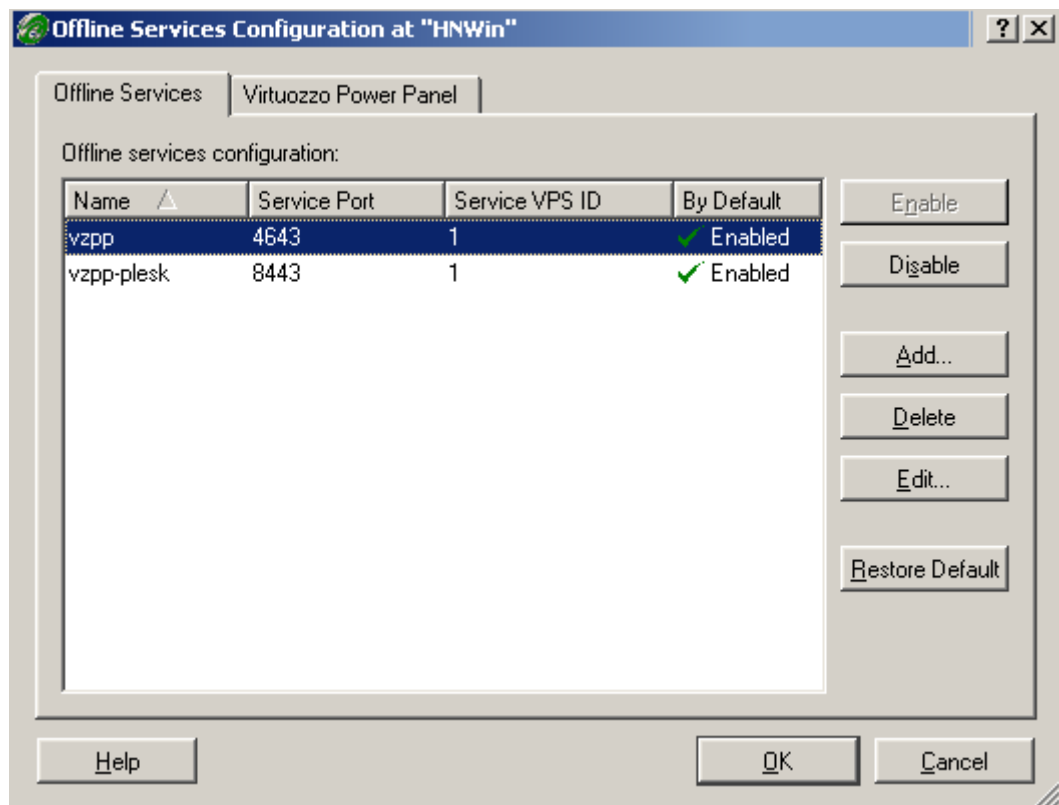


Figure 56: VZMC - Managing Offline Services Configuration

The information on offline services is presented in the table having the following columns:

Column Name	Description
Name	The name of the offline service to be used for managing VPSs by means of a standard web browser. The currently supported services are vzpp and vzpp-plesk.
Service Port	The communication port to be used by the offline service to access the VPS. The default port for the vzpp service is 4643; for the vzpp-plesk service - 8483.

Service VPS ID	The ID of the VPS where the request coming to the specified port are to be redirected. The requests for standard offline services (vzpp and vzpp-plesk) are redirected to VPS 1, i.e. to the Service VPS, responsible for managing all Virtual Private Servers on the Node. Then, these requests are transmitted to the corresponding VPS.
By Default	Indicates the status of the offline service: enabled or disabled. The status can be changed by using the Enable/Disable buttons to the right of the table.

On the **Offline Services** tab, you can perform the following operations:

- Press the **Add** button to the right of the **Offline Services** table to add a new offline service to be used for managing Virtual Private Servers thru a standard web browser. In the **New Offline Service** window displayed after clicking on the **Add** button, you should enter the name of the offline service, the communication port that will be used by the service to access the VPS, and the ID of the VPS where the requests coming to the specified port are to be redirected. The newly added offline service is disabled by default; to enable it, you should select the **Make default for all VPSs** checkbox.
- Use the **Delete** button to the right of the table to delete the selected offline service.
- Use the **Enable/Disable** buttons to the right of the table to enable/disable an existing offline service for all Virtual Private Servers residing on the Hardware Node.
- Select an offline service in the **Offline services configuration** table and click the **Edit** button to display the **Offline Service Properties** window, where you can change the name of the offline service, its communication port, and the ID of the VPS where the requests coming to the specified port are to be redirected. The newly added offline service is disabled by default; to enable it, you should select the **Make default for all VPSs** checkbox.
- Use the **Restore Default** button to the right of the table to restore the default offline services (i.e. vzpp and vzpp-plesk) with their default values.

Besides, you can globally enable/disable VZCC and VZPP to be used for managing Virtual Private Servers thru a standard web browser. To this effect, you should go to the **Virtuozzo Power Panels** tab and select/clear the **Enable Virtuozzo Power Panels and Control Center services** checkbox, respectively.

Viewing Summary Pages

You can view the summary page for every Hardware Node. Click on the name of the Hardware Node you are interested in in the tree in the left pane of the VZMC main window or double-click the name of the Hardware Node in the list of Nodes in the right pane.

The upper part of the information pane contains shortcuts to the most important tasks you are likely to do. However, all the actions and operations are accessible via the VZMC toolbar, **Action** menu and context menus. The bottom part of the Hardware Node summary page includes three tabs: **System**, **Network**, and **Disks**. The **System** tab describes the OS distribution and kernel version, CPU(s), RAM, and swap information. The **Network** tab describes the Hardware Node network configuration: interfaces and IP addresses. The **Disks** tab describes available disks and their utilization.

You can also view summary pages for each and every Virtual Private Server. To open the summary page in the Virtual Private Server Manager, click on the name of the Virtual Private Server in the tree pane. The summary page is similar to that in the main VZMC window:

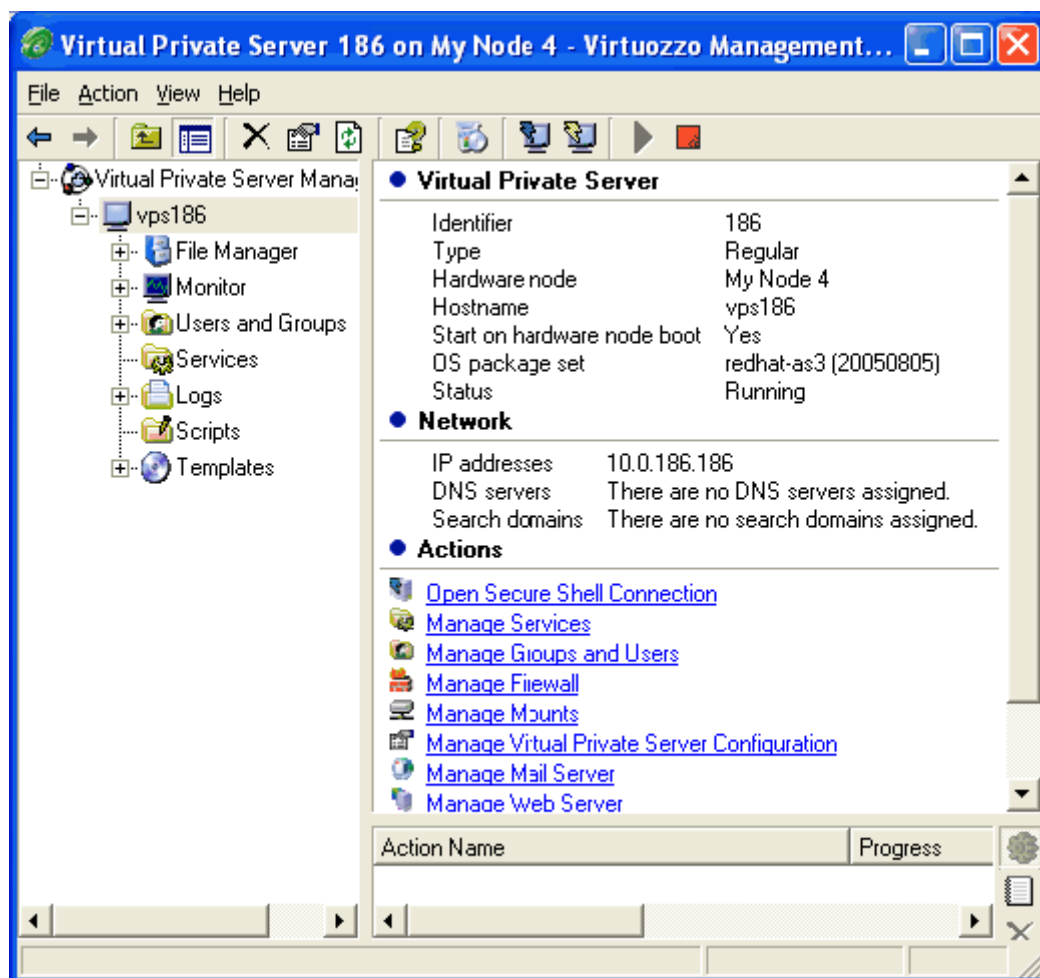


Figure 57: VZMC - Viewing VPS Summary Page

It contains information about VPS ID, type of the VPS, OS template, status (e. g. Mounted, Running), VPS class, and hostname. There is also a **Network** section describing the network configuration of the Virtual Private Server.

The shortcuts to the most common operations are located at the bottom of the summary page, in the **Actions** section.

Managing Users and Groups

Virtuozzo Management Console does not allow you to manage users or groups of the Host OS not to compromise the security of the Hardware Node. However, you can manage users and groups inside regular Virtual Private Servers with the help of Virtual Private Server Manager. All users and groups are adjustable. You can also add new users and groups.

To manage groups or users inside a Virtual Private Server, open the main tree for this Virtual Private Server and select either the **Groups** or **Users** items in the **Users and Groups** folder, respectively:

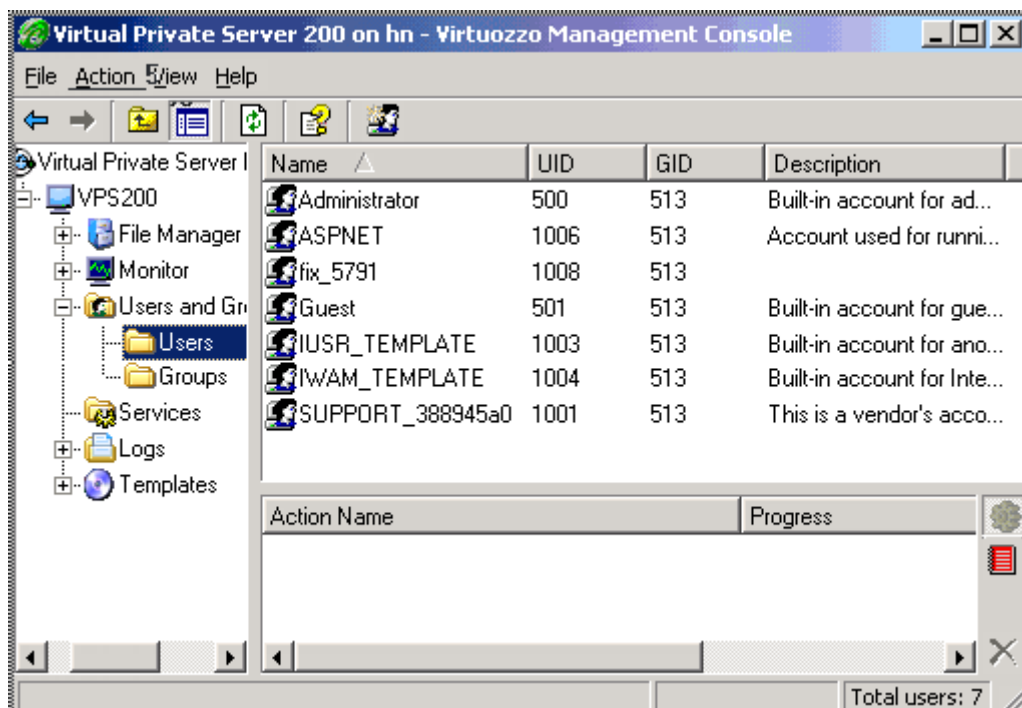


Figure 58: VZMC - Managing Users and Groups

To open the group properties dialog, double-click on the group name in the table of groups or select **Properties** on the context menu. To add a new user to the group, click the **Add** button. To remove a user from the group, select the user name and press the **Remove** button.

To add a new group, click the **New group** button on the toolbar (note that this button appears only if you are currently working with VPS groups). Then enter the group name and press **OK**.

To delete a group, select its name in the table of groups and press the **Delete** button on the toolbar or select the **Delete** item from the context menu.

To add a new user, open the list of users and click the **New user** button at the top toolbar. Enter the user login (user name). This is the only mandatory parameter. You may also specify the home directory, the login shell, set the user description and password, add the user to one or more groups (see the **Member Of** tab). Then press **OK**.

To edit an existing user, double-click on the user name in the table of users or use the **Properties** item from the context menu. The user properties dialog is analogous to the **New User** dialog.

To delete a user, select its name in the table of users and click the **Delete** button at the top toolbar or select the **Delete** option in the context menu.

Configuring Firewall

You can limit access of Internet users to your Hardware Node. To enable the Hardware Node firewall, click on the **Manage Hardware Node Firewall** link on the summary page, then mark the **Enable firewall protection** checkbox in the **Hardware Node Firewall Properties** dialog.

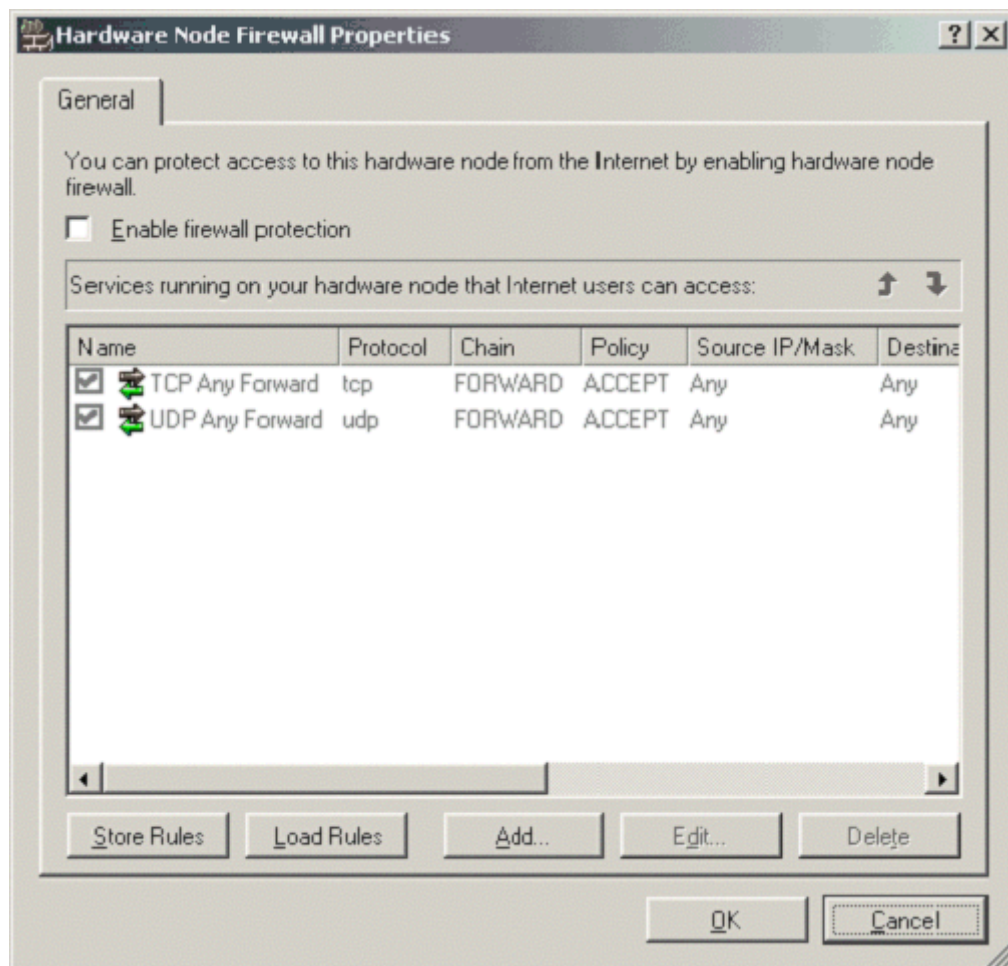


Figure 59: VZMC - Firewall Configuration Dialog

Two default rules are set for the Hardware Node, which are read-only. These rules are used to allow all Virtual Private Servers residing on the Node to receive/send IP packets from/to different networks via TCP and UDP protocols. You can add your own rules with the **Add** button, for example, to provide access to certain services like SSH, Telnet, POP3, SMTP, HTTP, and FTP. You can also define rules that are more specific. Refer to your Linux documentation for more details on firewall configuration.

You may remove rules from the existing list with the **Delete** button. To disable the rule temporarily, unmark the checkbox opposite the rule name. To change the rule, use the **Edit** button.

If you wish to enable firewall protection, but you are not sure which rules to use, you may follow these steps:

- 1** Right-click the name of the Hardware Node in the left pane of the VZMC window and select **Properties**;
- 2** On the **General** tab of the **Hardware Node Properties** window, press the **Configure** button in the **Securing hardware node** group;
- 3** Press **OK** in the pop-up window;
- 4** Mark the checkboxes beside those physical interfaces of the Hardware Node that you wish to protect;
- 5** Decide whether you wish to allow `ssh` connections to the Node, so select or clear the **Allow SSH connections to secured interfaces** checkbox;
- 6** Press **OK**.

Managing the firewall configuration for a Virtual Private Server is identical to managing the firewall configuration for the Hardware Node in respect of adding or removing rules. To manage the firewall configuration for a VPS, click the **Manage Firewall** link on the summary page of the Virtual Private Server Manager.

Each IP packet coming to a particular VPS passes 2 firewalls: the `iptables` rules of the Host OS and the firewall rules of the given VPS. An administrator of the Hardware Node sets up the Host OS `iptables` rules, and the end-users have no access to these rules.

Through the `venet` interface, packets reach the Virtual Private Server. The decision-making process is pretty similar to usual; however, the routing table in the VPS is different (the user can control it with the standard `iproute` package). The VPS also has the `INPUT`, `OUTPUT`, and `FORWARD` chains that can be used to control the traffic going in and out of the Virtual Private Server.

Managing Mount Points

You can manage mount points thru VZMC both for the Hardware Node and for each and every Virtual Private Server. To view the current list of mount points, click on the **Manage Mounts** link on the summary page of either the Hardware Node or the necessary VPS. Then use the **Add** button to add a new mount point, the **Delete** button to delete an existing mount point, or the **Edit** button to change an existing mount point:

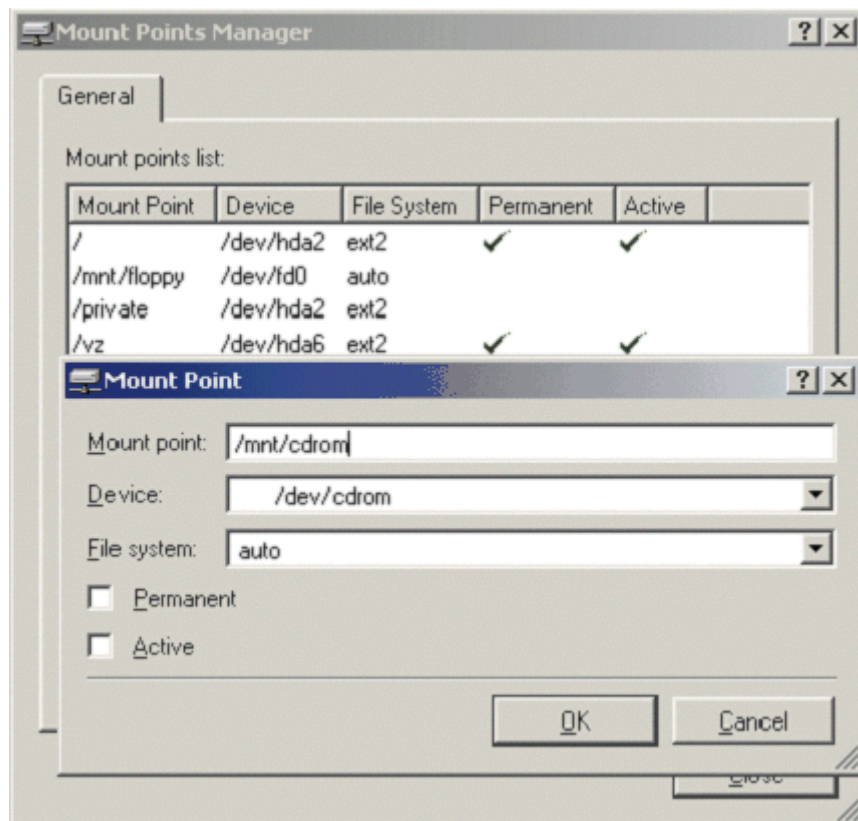


Figure 60: VZMC - Managing Mount Points

If you mark a mount point permanent, it means that this mount point will be automatically mounted on system boot. If you mark a mount point active, it will be mounted after you press the **OK** button. You may leave both checkboxes blank to create a mount point that is to be easily made active thru VZMC at the time the administrator chooses.

Viewing System and Virtuoizzo Logs

VZMC allows to view the logs which are maintained on the corresponding Hardware Node both for the Hardware Node itself and for a particular Virtual Private Server. The following log types are available for a particular Hardware Node in the VZMC main window:

Log type	Description
Alerts	Resource management system messages generated in case a VPS exceeds its Quality-of-Service limits or disk quotas.
Events	All VPS-related events (start, stop, migrate, mount, unmount, etc.).
Operations	Asynchronous tasks performed with any VPS of the Hardware Node.
Virtuoizzo	Full Virtuoizzo chronicles, i.e., system messages.
Actions	All actions performed with the main Virtuoizzo VPS management utility <code>vzctl</code> : creating a new Virtual Private Server destroying an existing Virtual Private Server, starting and stopping a Virtual Private Server, running commands in a Virtual Private Server and adjusting the configuration parameters and limits for a Virtual Private Server.

For Virtual Private Servers, only the **Alerts**, **Events**, and **Operations** logs together with the **Web** group of logs are available in the corresponding VPS manager window.

In order to view the logs, do the following:

- 1 Open the **Logs** folder in the main tree under either the Hardware Node name or the Virtual Private Server name and click on the needed log type.
- 2 Specify the time period for which you would like to view the logs.
- 3 Click **Download** to display the list of log entries in the right pane of the window.

Note: You can adjust the level of logging verbosity by defining the `log_level` parameter (from 0 to 2) in the Virtuoizzo global configuration file (adjustable by selecting the **Configuration** item in the HN main tree).

The **Web** group of logs in the Virtual Private Server manager window (available only for Linux HNs) is used to view the statistics of the http server(s) operation. The **Statistics** subfolder is used to generate the statistics report on the basis of a log file, and the **Text** subfolder is used to view log files in the textual format. In order to use this feature, do the following:

- 1 Select the corresponding subfolder in the left pane of the VPS manager window.
- 2 Press the **Generate Web Statistics** or **View Log File** button in the toolbar or right-click in the right pane and select the corresponding command in the context menu.
- 3 Select the necessary log file of the Virtual Private Server in the **Open** window that is displayed. You can choose any log file in any directory for viewing it in VZMC. For generating Web statistics, it is recommended to use the corresponding http server log files. For example, the statistics for VPS virtual hosts can often be generated on the basis of log files located in the `/home/<user_id>/<server_id>` directory.

Adjusting Template Configuration Files

The template configuration files determine the way the templates which have already been installed on the Hardware Node are to be added to Virtual Private Servers. For example, you can enable or disable preinstall and postinstall scripts, adjust the set of RPMs, etc. You can learn about the structure and the meaning of all the sections of template configuration files from the **Writing Template Configuration File** section of the **Managing Templates** chapter. While you are able to manually edit these files, you can also do it by means of VZMC. Do the following to view and/or edit a template configuration file in VZMC:

- 1 Open the **Templates** folder under the corresponding Hardware Node name in the VZMC main tree.
- 2 Click on either the **Application Templates** or **OS Templates** item to view the list of all application or OS templates, respectively.
- 3 Double-click on the name of the template to see the list of its versions.
- 4 Right-click on the needed version of the template (often there is only one version) and select **Properties** from the context menu (alternatively, you can select the version and click the **Properties** button at the top toolbar).
- 5 Go to one of the tabs and modify the template properties as needed.

The **Template Properties** window includes the following tabs:

Tab	Description
General	Here you can only add the description of the template. On a Windows Node, you can only view the description of the template.
Other	This is a read-only tab informing you of the IDs of those Virtual Private Servers where the given template has been added.
Actions	Here you can enable or disable preinstall and/or postinstall scripts. Just paste the script text in the appropriate textboxes or leave them empty.
Parameters	Here you can tune the package manager (which will add the specified command line switches passed to RPM during the template installation, uninstallation, and update) and other parameters added in the same way as during the template creation (see the Managing Templates chapter for more detail).
Packages	Here you can modify the set of RPMs included in the template. Packages that are currently included in the template are indicated with a tick. To detach a particular RPM from the template, unmark this RPM (click to remove the tick). Similarly, you can attach RPMs that were detached before. Obviously, you cannot just add RPMs to a template (to this effect, you need to create a template update). The read-only list of packages that should be installed without resolving dependencies and without executing scripts is displayed at the bottom part of the dialog window. Practically any OS template requires such a list to "bootstrap" the basic system before installing the main set of packages.

Sets This tab is present in the **Template Properties** window of application templates only. You can add or remove the names of the OS templates the given application template must be compatible with. By including OS templates to this list, you make the program not check up the templates compatibility when adding the application template to a VPS based on the given OS template. That is why, you should be sure the templates are compatible with each other.

Managing Files

You cannot manage files directly on the Hardware Node by means of VZMC, but you can do it inside each and every Virtual Private Server by means of the Virtual Private Server manager window. After you click on the **File Manager** item in the VPS main tree, you will see the list of folders and files of the VPS root directory. Thus, this item corresponds to the `/` directory of the selected VPS:

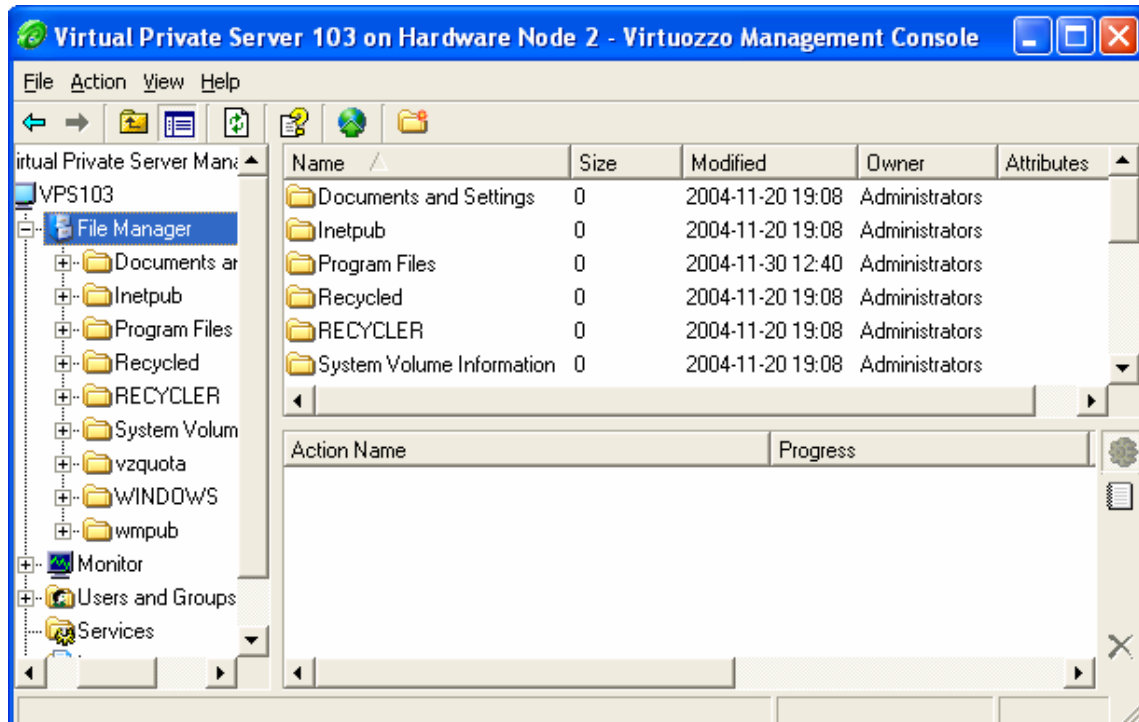


Figure 61: VZMC - Managing Files

The principles of working with the VPS file manager are standard. You can move thru the hierarchy of VPS folders by double-clicking the folders names or selecting the necessary folders in the left pane. Use the menu items, toolbar buttons, table view, and context menus to perform the following tasks:

- View the contents of simple text files;
- View the principal information about a file/folder/symlink located in every directory and subdirectory of any depth in the given VPS;
- Upload any number of files or whole directories from the local computer (the computer where VZMC is installed) to any folder of the given VPS;
- Download any number of files from the given VPS to the local computer;
- Create new folders in the VPS;
- Copy files to another directory in the given VPS;
- Move files to another directory in the given VPS;
- Delete VPS files;
- Rename VPS files;

- Set permissions for VPS files.

VZMC provides a user-intuitive interface for performing all these tasks.

Managing Search Domains

Search domains is the list for hostname lookup. The search list is normally determined by the local domain name; by default, it contains only the local domain name. You can add other host names for a particular VPS. A search query is performed by attempting to use each item in the list in turn until a match is found. Note that this process may be slow and may generate a lot of network traffic if the servers for the listed domains are not local, and that the query might time out if no server is available for one of the domains. The search list is currently limited to six domains with a total of 256 characters.

To view and/or edit the list of search domains for a particular VPS, do the following:

- 1 Click on the **Virtual Private Servers** item in the VZMC main tree.
- 2 As soon as the list of the VPSs on this particular Hardware Node is displayed, right-click on the necessary VPS name and select **Properties** on the context menu. (In case you are working with the Virtual Private Server Manager, click on the **Manage Virtual Private Server Configuration** link at the VPS dashboard).
- 3 Click the **Network** tab in the **Virtual Private Server Properties** window.
- 4 Under the **Search domains** group, use the **Add search domain**, **Delete search domain**, **Search domain property** buttons to add, delete, or edit a search domain, respectively.

Managing Virtual Private Server Servers

Many OS templates have different kinds of servers installed by default, including mail servers, web servers, and FTP servers. VZMC provides a convenient means for managing the main parameters of these servers. It can work with the standard Linux mail packages, the Apache web server, and the ProFTPD server. If any of these servers is missing from a Virtual Private Server, it can be installed by creating the corresponding template, installing it and adding to this Virtual Private Server (see the **Managing Templates** chapter for details).

To configure a VPS server, open the corresponding server configuration window by clicking the **Manage Mail Server**, **Manage Web Server**, or **Manage FTP Server** link at the VPS dashboard, or right-click the name of the Virtual Private Server in the VPS manager window, point to **Control Panels** and choose the analogous options.

The instructions on working with VPS Control Panels are provided in the VZMC online context help.

Checking Virtual Private Server

This feature uses a number of tests for troubleshooting the problem Virtual Private Servers that may not start, are not accessible from the network, or run with low performance. To run a set of such tests (or a *test script*), you should launch the **Check Virtual Private Server** wizard by right-clicking the VPS in question and selecting **Check Virtual Private Server --> Tasks** on the context menu.

On the first page of the wizard (**Choose Test Script to Run**), you may define the problem with your VPS for the wizard to offer the most suitable set of scripts. This page is only used for the wizard to make such a suggestion, you may customize the test script to run on the next page whatever this suggestion.

The second page of the wizard (**Choose Tests to Run**) lists all available tests grouped by categories in the left pane, from which you may add them to a test script to be performed with the help of the **Add** button. If you chose a VPS problem on the previous page, the relevant tests are already selected for the script, though you may still customize it at your own discretion. The toolbar buttons allow you to save and load test scripts, to alter the order of performing the tests and to receive extra information on the selected tests. Press **Next** to begin running the test script.

The third and the fourth pages of the wizard present you with the report on the tests results and provide recommendations on the recovery of the Virtual Private Server.

CHAPTER 14

Troubleshooting

This chapter provides the information about those problems that may occur during your work with Virtuozzo and suggests the ways to solve them, including getting technical support from SWsoft.

In This Chapter

General Considerations	292
Kernel Troubleshooting	294
Problems With VPS Management	296
Problems With VPS Operation	300
Problems With Physical Server Migration.....	300
Miscellaneous Problems	301
Getting Technical Support	302

General Considerations

The general issues to take into consideration when troubleshooting your Virtuozzo system are listed below. You should read them carefully before trying to solve more specific problems.

- Make sure a valid license is always loaded on the Hardware Node. If your license has expired and the grace period is over, all the VPSs on your Node will be stopped!
- You should always remember where you are located now in your terminal. Check it periodically using the `pwd`, `hostname`, `ifconfig`, `cat /proc/vz/veinfo` commands. One and the same command executed inside a VPS and at the HN can lead to very different results! You can also set up the `PS1` environment variable to show the full path in the `bash` prompt. To do that, add these lines to `/root/.bash_profile`:

```
PS1="[ \u@\h \w]$ "
export PS1
```

- If the Hardware Node slows down, use `vmstat`, `ps (ps axfw)`, `dmesg`, `top (vztop)` to find out what is happening, never reboot the machine without investigation. If no thinking helps restore the normal operation, use the `Alt+SysRq` sequences to dump the memory (`showMem`) and processes (`showPc`).
- If the Hardware Node was incorrectly brought down, on its next startup all the partitions will be checked and quota recalculated for each VPS, which dramatically increases the startup time.
- Do not run any binary or script that belongs to a VPS directly from the Hardware Node, for example, do not ever do that:

```
cd /vz/root/99/etc/init.d
./httpd status
```

Any script inside a VPS could have been changed to whatever the VPS owner chooses: it could have been trojaned, replaced to something like `rm -rf`, etc. You can use only `vzctl exec/vzctl enter` to execute programs inside a VPS.

- Do not use `init` scripts at the Hardware Node. An `init` script may use `killall` to stop a service, which means that all similar processes will be killed in all VPSs! You can check `/var/run/Service.pid` and kill the correspondent process explicitly.
- You must be able to detect any rootkit inside a VPS. It is recommended to use the `chkrootkit` package for detection (you can download the latest version from www.chkrootkit.org), or at least run

```
rpm -Va | grep "S.5"
```

to check up if the MD5 sum has changed for any RPM file.

You can also run `nmap`, for example:

```
# nmap -p 1-65535 192.168.0.1

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.0.1):
(The 65531 ports scanned but not shown below are in
state: closed)
Port      State      Service
21/tcp    open       ftp
22/tcp    open       ssh
80/tcp    open       http
```

```
111/tcp    open      sunrpc
```

```
Nmap run completed -- 1 IP address (1 host up) scanned  
in 169 seconds
```

to check if any ports are open that should normally be closed.

That could however be a problem to remove a rootkit from a VPS and make sure it is 100% removed. If you're not sure, create a new VPS for that customer and migrate his/her sites and mail there.

- Check the `/var/log/` directory on the Hardware Node to find out what is happening on the system. There are a number of log files that are maintained by the system and Virtuozzo (the `boot.log`, `messages`, `vzagent.log` log files, etc.), but other services and programs may also put their own log files here depending on your distribution of Linux and the services and applications that you are running. For example, there may be logs associated with running a mail server (the `maillog` file), automatic tasks (the `cron` file), and others. However, the first place to look into when you are troubleshooting is the `/var/log/messages` log file. It contains the boot messages when the system came up as well as other status messages as the system runs. Errors with I/O, networking, and other general system errors are reported in this file. So, we recommend that you turn to the `messages` log file first and then proceed with the other files from the `/var/log/` directory.
- Subscribe to bug tracking lists, at least for Red Hat. You should keep track of new public DoS tools or remote exploits for the software and install them into VPSs or at Hardware Nodes.
- When using `iptables`, there is a simple rule for Chains usage to help protect both the HN and its VPSs:
 - use `INPUT`, `OUTPUT` to filter packets that come in/out the HN;
 - use `FORWARD` to filter packets that are designated for VPSs.

Kernel Troubleshooting

Using ALT+SYSRQ Keyboard Sequences

Press ALT+SYSRQ+H (3 keys simultaneously) and check what's printed at the HN console, for example:

```
SysRq: unRaw Boot Sync Unmount showPc showTasks showMem loglevel0-8
tErm kIll killall Calls Oops
```

This output shows you what ALT+SYSRQ sequences you may use for performing this or that command. The capital letters in the command names identify the sequence. Thus, if there are any troubles with the machine and you're about to reboot it, please press the following sequences before pressing the Power button:

ALT+SYSRQ+M to dump memory info;

ALT+SYSRQ+P to dump processes states;

ALT+SYSRQ+S to sync disks;

ALT+SYSRQ+U to unmount filesystems;

ALT+SYSRQ+L to kill all processes;

ALT+SYSRQ+U try to unmount once again;

ALT+SYSRQ+B to reboot.

If the computer is not rebooted after that, you can press the Power button.

Saving and Decoding Kernel Fault (OOPS)

You can use the following command to check for the kernel messages that should be decoded and reported to Virtuozzo developers:

```
grep -E "Call Trace|Code" /var/log/messages*
```

Then you should find these lines in the correspondent log file and figure out what kernel was booted when the oops occurred. Search backward for the "Linux" string, look for strings like that:

```
Jun 1 11:41:12 hsp-sup kernel: Linux \
version 2.4.1-008stab035.7.swsoft (st@rel202.eng.sw.ru) \
(gcc version 2.96 20000731 (Red Hat Linux 7.1 2.96-81)) \
#1 Wed Jun 26 21:34:13 MSD 2002
```

An oops usually starts with some description of what happened and ends with the Code string. Here is an example:

```
Feb 25 12:37:51 vz1 kernel: Unable to handle kernel NULL pointer \
dereference at virtual address 0000000d
Feb 25 12:37:51 vz1 kernel: printing eip:
Feb 25 12:37:51 vz1 kernel: c0127117
Feb 25 12:37:51 vz1 kernel: *pde = 1fc1f001
Feb 25 12:37:51 vz1 kernel: *pte = 00000000
Feb 25 12:37:51 vz1 kernel: Oops: 0000
Feb 25 12:37:51 vz1 kernel: CPU: 1
Feb 25 12:37:51 vz1 kernel: EIP: 0010:[<c0127117>] Tainted: P
Feb 25 12:37:51 vz1 kernel: EFLAGS: 00010202
Feb 25 12:37:51 vz1 kernel: eax: 00000001 ebx: ebb06624 \
ecx: 00000001 edx: ebb06500
Feb 25 12:37:51 vz1 kernel: esi: ec2c1224 edi: d14cbef8 \
ebp: 0804c9f4 esp: d14cbec4
Feb 25 12:37:51 vz1 kernel: ds: 0018 es: 0018 ss: 0018
Feb 25 12:37:51 vz1 kernel: Process ssl-gw (pid: 16089, \
stackpage=d14cb000)
Feb 25 12:37:51 vz1 kernel: Stack: ebb06624 ec2c1224 d14cbef8 c026ad3e
Feb 25 12:37:51 vz1 kernel: Call Trace: [<c026ad3e>] [<c026b9f3>] \
[<c026ba76>] [<c011550a>] [<c0114ead>]
Feb 25 12:37:51 vz1 kernel: [<c026c440>] [<c01071e3>]
Feb 25 12:37:51 vz1 kernel:
Feb 25 12:37:51 vz1 kernel: Code: 8b 41 ea 08 31 c2 81 e2 ff 00 00 00 \
c1 e2 02 8d
```

All you need is to put the oops into a file and then issue the following command, assuming that the file name is `1.oops`:

```
ksymoops -KVLO -m /boot/System.map-2.4.1-008stab043.15.swsoft <1.oops
```

`/boot/System.map-2.4.1-008stab043.15.swsoft` is a system map of the kernel on which this oops occurred. You can simply use

```
ksymoops -KVLO <1.oops
```

if the oops occurred on the kernel that is currently running.

Finding Kernel Function That Caused D Process State

If there are too many processes in the D state and you can't find out what is happening, issue the following command:

```
# objdump -Dr /boot/vmlinux-`uname -r` >/tmp/kernel.dump
```

and then get the process list:

```
# ps axfwln
  F  UID      PID  PPID  PRI  NI   VSZ   RSS   WCHAN  STAT  TTY  TIME  COMMAND
100   0    20418  20417   17   0  2588   684      -  R    ?   0:00  ps axfwln
100   0         1      0    8   0  1388   524 145186  S    ?   0:00  init
040   0     8670     1    9   0  1448   960 145186  S    ?   0:00  syslogd -m 0
040   0     8713     1   10   0  1616  1140 11ea02  S    ?   0:00  crond
```

Look for a number under the **WCHAN** column for the process in question. Then you should open `/tmp/kernel.dump` in an editor, find that number in the first column and then scroll backward to the first function name, which can look like this:

```
"c011e910 <sys_nanosleep>:"
```

Then you can tell if the process “lives” or is blocked into the found function.

Problems With VPS Management

This section includes recommendations on how to settle some problems with your VPSs.

Failure to Create VPS

An attempt to create a new Virtual Private Server fails. There is a message on the system console: `Cached package set XXX version YYY not found.`

Solution 1

The necessary OS template might be absent from the Hardware Node. Copy the template to the Hardware Node, install it, cache it, and try to create a VPS once again.

Solution 2

The VPS private area might not be pre-cached. In this case the `vzpkgcache` utility shall be used. Issue the command:

```
vzpkgcache
```

The utility looks for the OS templates installed on the Hardware Node and caches those that are not cached. After this, try to create a VPS once again.

Failure to Start VPS

An attempt to start a Virtual Private Server fails.

Solution 1

If there is a message on the system console: `parameters missing`, and the list of missed parameters follows the message, set these parameters using the `vzctl set --save` command (see [Configuring Virtual Private Server](#) on page 44 for instructions). Try to start the VPS once again.

Solution 2

If there is a message on the system console: `IP address is already used`, issue the `cat /proc/vz/veinfo` command. The information about the VPS numeric identifier, VPS class, number of VPS's processes and VPS IP address shall be displayed for each running VPS. This shall also demonstrate that your VPS is up, i.e. it must be running without any IP address assigned. Set its IP address using the command:

```
vzctl set ve_id --ipadd addr --save
```

where `ve_id` represents the VPS numeric identifier and `addr` represents an actual IP address.

Solution 3

Poor UBC parameters might prevent the VPS from starting. Try to validate the VPS configuration (see [Validating Virtual Private Server Configuration](#) on page 151). See what configuration parameters have caused the error and set appropriate values using the `vzctl set --save` command.

Solution 4

The VPS might have used all its disk quota (either disk space or disk inodes). Check the VPS disk quota (see the [Managing Disk Quotas](#) section and Chapter 7 for details) and increase the quota parameters if needed (see [Setting Up Per-VPS Disk Quota Parameters](#) on page 115).

Solution 5

Run the `vzfsutil` utility to make sure that the `vzfs` symlinks inside the VPS work correctly. For example:

```
vzfsutil --call -t /vz/template /vz/private/<VPS_ID>
```

The complete reference on the `vzfsutil` utility is provided in the [Virtuozzo Reference Guide](#).

Solution 6

The VPS administrator might have inadvertently modified, replaced, or deleted any file that is part of an application or OS template, which has brought about the VPS malfunction. In this case, restore the file(s) with the `vzctl recover` command (see the [Recovering Virtual Private Server](#) section for details).

Solution 7

Restore the latest operable copy of the VPS by means of the `vzrestore` utility (see the [Backing Up and Restoring Virtual Private Server](#) section for details).

Failure to Access VPS From Network

Solution 1

The IP address assigned to this Virtual Private Server might be already in use in your network. Make sure it is not. The problem VPS address can be checked by issuing the following command:

```
# grep IP_ADDRESS /etc/sysconfig/vz-scripts/<VE_ID>.conf
IP_ADDRESS="10.0.186.101"
```

The IP addresses of other VPSs, which are running, can be checked by running

```
cat /proc/vz/veinfo
```

Solution 2

Make sure the routing to the Virtual Private Server is properly configured. Virtual Private Servers can use the default router for your network, or you may configure the Hardware Node as router for its VPSs.

Failure to Log In to VPS

The Virtual Private Server starts successfully, but you cannot log in.

Solution 1

You are trying to connect via SSH, but access is denied. Probably you have not set the password of the root user yet or there is no such user. In this case, use the `vzctl set --save --userpasswd` command. For example, for Virtual Private Server 101 you might issue the following command:

```
# vzctl set 101 --save --userpasswd root:secret
```

Solution 2

Check forwarding setting by issuing the command:

```
# cat /proc/sys/ipv4/conf/venet0/forwarding
```

If it is 0 then change it to 1 by issuing the command:

```
# echo 1 > /proc/sys/ipv4/conf/venet0/forwarding
```

Failure to Back Up VPS in VZMC

An attempt to back up a Virtual Private Server with a large amount of disk space (e.g. 6 Gb) by means of VZMC finishes with the following error message: The request was timed out. However, the backup process continues running and the VPS backup is successfully created on the Backup Node after a while, which can be checked by exploring the `/vz/backup` directory on this Node, where all VPS backups are stored by default.

Solution

The problem is caused by the fact that the timeout limit set by VZAgent for the VPS backup process in VZMC has been reached. This limit is equal to 3600 seconds by default. You can increase the maximal backup timeout value by performing the following operations:

- 1 In VZMC, right-click on the Hardware Node name and select **Tasks --> Manage VZAgent Configuration** on the context menu.
- 2 In the left part of the displayed window, choose **backm --> configuration --> timeouts**.
- 3 Double-click the `backup` parameter in the right part of the **VZAgent Configuration** window and specify the needed time (in seconds) in the **Parameter value** field.
- 4 Click OK.

Failure to Display List of VPS Backups

You created a number of VPS backups on the Backup Node and now wish to view them. However, the process of displaying your VPS backups takes a very long time or even goes into infinity.

Solution

By default, the timeout limit for the VPS backup search process is set to a very high value - 3600 seconds, which makes the search process to run for 60 minutes before showing a list of available backups on the Backup Node. To reduce the time needed to display your VPS backup list, you should decrease the backup search value. You can do it in the following way:

- 1 In VZMC, right-click on the Hardware Node name and select **Tasks --> Manage VZAgent Configuration** on the context menu.
- 2 In the left part of the displayed window, choose **backm --> configuration --> timeouts**.
- 3 Double-click the `search` parameter in the right part of the **VZAgent Configuration** window and specify the desired time (in seconds) in the **Parameter value** field.

Note: You are recommended to set the value of the `search` parameter to 300 seconds.

- 4 Click OK.

Problems With VPS Operation

Timeout When Accessing Remote Hosts

A host is unreachable by the Virtuozzo Hardware Node or its Private Servers, though it can be reached from other computers.

Solution 1

Often these timeouts occur due to the fact that the Explicit Congestion Notification (ECN) mechanism of the TCP/IP protocol is on by default in Virtuozzo and off in some other systems, which leads to their incompatibility. ECN is used to avoid unnecessary packet drops and for some other enhancements. If Virtuozzo cannot connect to a host, turn off this mechanism:

```
# sysctl -w net.ipv4.tcp_ecn=0
net.ipv4.tcp_ecn = 0
```

Extraneous Backups Visible to VPS in VZPP

Sometimes the Back Up/Restore VPS page in VZPP shows backups not belonging to the given VPS.

Solution 1

This happens when two or more Hardware Nodes have VPSs with identical IDs hosted on them. If such VPSs are backed up onto one and the same Backup Node, they will be able to see the backups of each other by means of VZPP. To avoid this situation, you are recommended to have unique VPS IDs throughout all your Hardware Nodes. Consider using the *Virtuozzo Cluster* feature of VZMC to facilitate performing this task.

Problems With Physical Server Migration

Failure to Start iptables Modules After Physical Server Migration

iptables is broken in the VPS after a physical server has been migrated.

Solution

The iptables service can work properly inside the VPS that has resulted from a physical server migration only if the `ipt_state` module is loaded both on the Hardware Node and in the VPS in question. The simplest way to do it is the following:

- 1 Stop Virtuozzo on the Node:

```
# service vz stop
```

2 Add `ipt_state` as another module name to the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file on the Node.

3 Restart `iptables` on the Node:

```
service iptables restart
```

4 Start `Virtuozzo`:

```
# service vz start
```

5 Add `ipt_state` as another module name to the `IPTABLES` parameter in the `/etc/sysconfig/vz` file on the Node.

6 Restart the VPS:

```
# vzctl restart VPS_ID
```

To learn more on loading `iptables` modules, please turn to the [Loading iptables Modules](#) section (on page 253).

Miscellaneous Problems

Failure to Run `vgscan` Utility

The `vgscan` utility may not work on some Host operating systems using `lvm2`-compatible tools (for example, Fedora Core 2), if `Virtuozzo` is installed on such systems. To avoid the utility malfunction, make sure you have the `lvm1`-compatible tools installed on your Node.

Corrupted Pseudographics in `Virtuozzo` Utilities

Some `Virtuozzo` utilities (e.g. `install`, `vzup2date`, and others) employ pseudographical instead of simple character output during their operation. Certain terminal clients fail to display the pseudographics the way it was intended to be displayed. This has nothing to do with `Virtuozzo`, but with locale settings either on the Hardware Node or in the terminal client. You may try to solve this problem in one of the following ways:

Solution 1

Set the correct locale for your terminal.

Solution 2

Try to run the utility as

```
# LC_ALL=C utility_name
```

Solution 3

If you are connecting to the Node via a remote shell, please make sure the locale set in the remote terminal is the same as in the local one.

Getting Technical Support

Getting Assistance With Virtuozzo Installation

SWsoft provides installation assistance for the Virtuozzo software. Assistance with installation can be offered via e-mail or by using the *Virtuozzo Support Tunnel* tool:

- While communicating via e-mail, the SWsoft support will attempt to answer any relevant questions you may have before the installation process is initiated. This includes the following:
 - Pre-requisites list;
 - Hardware compatibility;
 - Software compatibility.
- You can also install the *Virtuozzo Support Tunnel* tool on your physical computer and use it for getting installation assistance from the SWsoft support. Detailed information on the *Virtuozzo Support Tunnel* tool is provided in the *Establishing Secure Channel to SWsoft Support* subsection (on page 305).

Preparing and Sending Questions to Technical Support

In most cases, the support team must rely on the customer's observations and communications with the customer in order to diagnose and solve the problem. Therefore, the detailed problem report is extremely important. Please, do mention the following:

- Symptoms of the problem;
- When the problem began including the circumstances of the failure;
- Any changes you made to your system;
- Other information that may be relevant to your situation, such as the installation method;
- Specific hardware devices that may be relevant to your problem.

After you have prepared your questions, you can send them to the support team to support@swsoft.com.

You can also call us or visit one of our offices. Information about phone numbers, contact people and office addresses is available on the contact page at <http://www.swsoft.com/en/contact>.

Submitting Problem Report to Technical Support

Virtuozzo 3.0 is shipped with a special utility - `vzreport` - allowing you to compile a detailed report, in case you have any Virtuozzo-related problems, and to automatically send it to the SWsoft support team.

`vzreport` has two modes of execution — full screen and command line. By default, the utility starts in the full screen mode; however, you can force it to run in the command line mode by specifying any option containing your contact information (e.g. `-n` denoting your name) or the problem report description (e.g. `-m` used to provide additional information on your problem). Detailed information on all the options that can be passed to `vzreport` in the command line is provided in the [Virtuozzo Reference Guide](#).

After running the `vzreport` utility in the full screen mode (i.e. without specifying any option), the Problem Report Wizard is opened, which will guide you thru a number of steps asking you to provide the necessary information to generate a problem report. On the **Welcome to ...** screen, just click **Next** to proceed with the wizard. You will be presented with the following window:

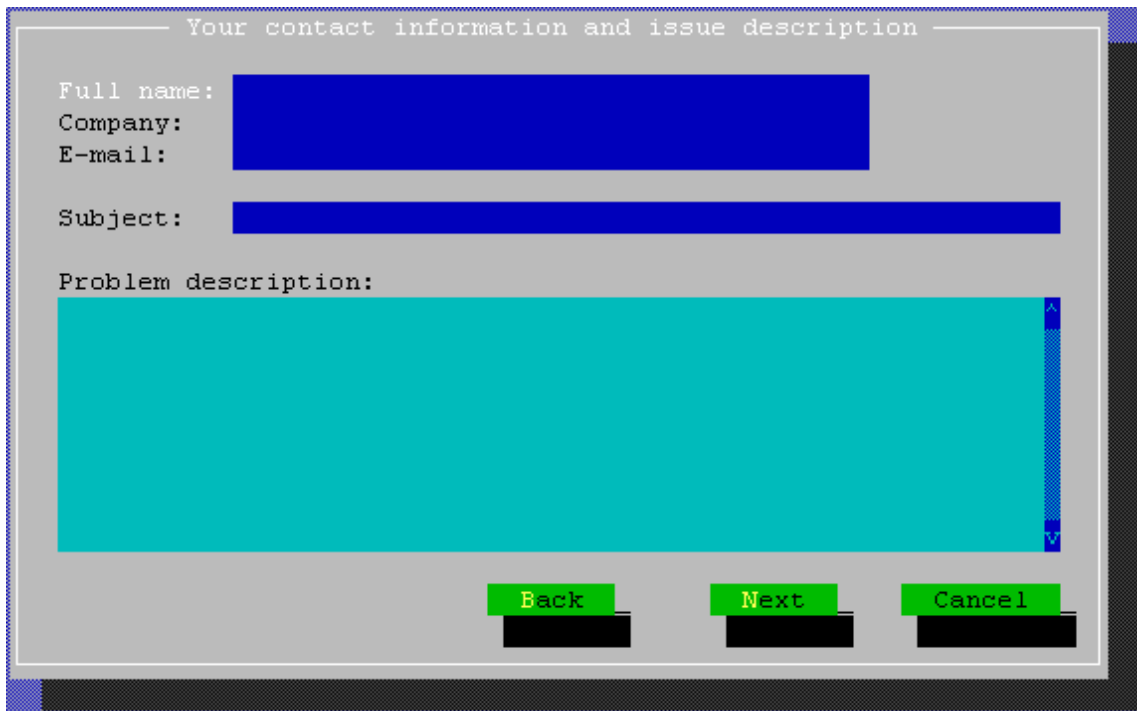


Figure 62: Submitting Problem Report - Entering name and e-mail address

In this window, you should enter your name, e-mail address, and the name of your company into the corresponding fields. Make sure that you type a valid e-mail address; otherwise, the SWsoft support team will not be able to contact you. In the **Subject** field, you should also specify what Virtuozzo problem you encountered and may provide additional information in the **Problem description** field which, in your opinion, can help solve the problem.

Clicking **Next** in the **Your contact information and issue description** window starts collecting Virtuozzo logs and information on your system and network settings into a special file. This file will be sent to the SWsoft support team upon the completion of the wizard. The file does not contain any private information!

After the utility has gathered all the necessary information on your Node, the **Submit report** window is displayed:

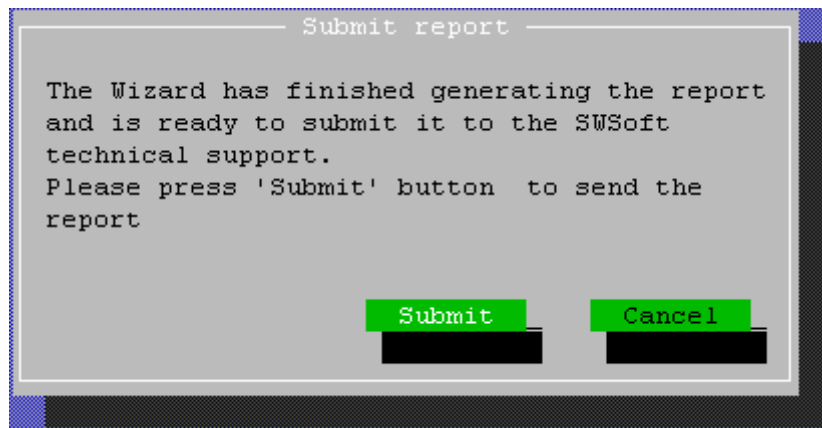


Figure 63: Submitting Problem Report - Sending Report to SWsoft

Press the **Submit** button in this window to send your problem report to the SWsoft technical support team. The report is dispatched directly to SWsoft by using the HTTP protocol and port 80. However, if you use an HTTP proxy server for handling all your HTTP request and wish your problem reports to be sent via this server, you should specify the hostname or IP address of the server in the `/etc/sysconfig/vz` configuration file on the Hardware Node as the value of the `HTTP_PROXY` parameter.

After the problem report has been successfully sent to the SWsoft support, the **Congratulations** window is displayed informing you of the process completion and the ID assigned to the report. You can use this ID later on while contacting the SWsoft support to uniquely identify your report among the other ones.

Note: There may happen situations when the problem report cannot be automatically sent to the SWsoft support for some reason (e.g. problems with the Internet connection). In this case the corresponding warning message will be displayed telling you the path to the file on the Node where the generated message is located. Please send this file manually to the support team at support@swsoft.com.

Establishing Secure Channel to SWsoft Support

Virtuozzo provides you with a special tool - *Virtuozzo Support Tunnel* - which allows you to establish a private secure channel to the SWsoft support team server. After establishing such a channel, the support team will be able to quickly and securely connect to your Node and diagnose and solve your problem. The secure connection to your computer is achieved thru a Virtual Private Network (VPN) created between the SWsoft support team server and your Hardware Node.

To start using the *Virtuozzo Support Tunnel* tool, you should:

- Make sure the `openvpn` (version 2.0 and above) and `vzvpn` packages are installed on your Node. These packages are automatically installed on the Node during the installation of Virtuozzo version 2.6.2 to 3.0. However, if you are running a Virtuozzo version older than 2.6.2, you may need to manually copy these packages and install them on your Node.
- Ask SWsoft for a special certificate which will uniquely identify you as a Virtuozzo user. Certificates are issued by SWsoft in the form of files and should be installed on your Node by issuing the `vzvpn.sh key-install certificate` command where *certificate* denotes the name of the certificate file obtained from SWsoft.

After you have completed the tasks above and in case you encountered a Virtuozzo-related problem, you can do the following to get assistance from the SWsoft support:

- 1 Make sure your Hardware Node is connected to the Internet.
- 2 On the Node, execute the `/etc/init.d/vzvpn start` command to establish a VPN between your Node and the SWsoft support server.
- 3 Contact the SWsoft support team (by telephone or via e-mail) and inform them of the problem you encountered. You should also mention that you have launched the *Virtuozzo Support Tunnel* tool and established a VPN to the SWsoft support server.
- 4 After that, the SWsoft support team will closely examine your problem, connect to your Node by using the secure VPN established, if needed, and make its best to solve your problem as quickly as possible.

Note: *Virtuozzo Support Tunnel* is implemented as a standard Linux service running in the background of your system. Therefore, to have this service running after your Hardware Node reboot, you should set it to the `autoboot` mode or start it manually again by executing the `/etc/init.d/vzvpn start` command.

Glossary

Application template is a template used to install a set of applications in *Virtual Private Servers*. See also *Template*.

EZ template is a template file that points to a repository with the packages that comprise the template. Unlike *standard templates*, EZ templates cannot be updated because the repository stays the same. However, the packages in the repository can be updated.

Hardware Node (or *Node*) is a computer where Virtuozzo is installed for hosting *Virtual Private Servers*. Sometimes, it is marked as *VPS 0*.

HN is an abbreviation of *Hardware Node*.

Host Operating System (or *Host OS*) is an operating system installed on the *Hardware Node*.

HSPcomplete is an end-to-end solution for Hosting Service Providers by SWsoft. Is dependent on *Virtuozzo*.

MAC address stands for Media Access Control address, a hardware address that uniquely identifies each Node in a network. The MAC layer interfaces directly with the network media. Consequently, each different type of network media requires a different MAC layer.

Management Node is a computer used in the *HSPcomplete* solution to control the *Hardware Nodes*. It runs services for a web-based management interface and database containing information about customers and services.

mlock, *mlock'ed page* — `mlock()` (short for memory locking) is a system call. It disables paging out for a specific region of memory. `mlock'ed` pages are guaranteed to stay resident in RAM until they are unlocked by `munlock()` system call. There are two primary applications of memory locking — the real-time applications and high-security data processing. The former require the deterministic response. The latter needs to protect valuable data from paging out into the swap file.

MN is an abbreviation of *Management Node*.

OS template (or *Operating System template*) is used to create new *Virtual Private Servers* with a preinstalled operating system. See also *Template*.

Package set is a synonym for *Template*.

Private area is a part of the file system where *VPS* files that are not shared with other *Virtual Private Servers* are stored.

QoS is an abbreviation of *Quality of Service*.

Quality of Service specifies the performance properties of a *VPS*. It allows you to provide a fair share of the main system resources among *Virtual Private Servers* and better service quality for preferred *VPSs*.

siginfo structure (or just *siginfo*) is a block of information about signal generation. If a process catches a signal, it may receive *siginfo* telling why the system generated that signal. If a process monitors its children, it may receive *siginfo* telling why a child has changed its state. In either case, the system returns the information in a structure of the *siginfo_t* type, which includes the following information: signal number, error number, and signal code.

SSH stands for Secure Shell. It is a protocol for logging on to a remote machine and executing commands on that machine. It provides secure encrypted communications between two untrusted hosts over an insecure network.

Service Virtual Private Server is a special VPS running *VZAgent* which is responsible for managing all the *Virtual Private Servers* of the given *Hardware Node*. You should use the IP address of the Service VPS to connect to a *Hardware Node* by means of *VZMC* or *VZCC*. The Service VPS is always marked as Virtual Private Server 1.

Service VPS is an abbreviation of *Service Virtual Private Server*.

Standard template is a template file that has inside itself all the re-usable files of all the packages comprising the template. If newer versions of any of these packages appear, a standard template can be correspondingly updated. Compare *EZ template*.

SVPS is an abbreviation of *Service Virtual Private Server*.

TCP (TCP/IP) stands for Transmission Control Protocol/Internet Protocol. This suite of communications protocols is used to connect hosts on the Internet.

Template (or *package set*) is a set of original application files (packages) repackaged for mounting over Virtuozzo File System. There are two types of templates. OS Templates are used to create new Virtual Private Servers with a preinstalled operating system. Application templates are used to install an application or a set of applications in *Virtual Private Servers*. See also *Standard template* and *EZ template*.

UBC is an abbreviation of *User Beancounter*.

User Beancounter is the subsystem of Virtuozzo for managing VPS memory and some system-related resources.

Virtual Environment (or *VE*) is an obsolete designation of a virtual private server. See *Virtual Private Server*.

VPS is an abbreviation of *Virtual Private Server*.

VENET device is a virtual networking device, a gateway from a *VPS* to the external network.

Virtual Private Server is a virtual private server, which is functionally identical to an isolated standalone server, with its own IP addresses, processes, files, its own users database, its own configuration files, its own applications, system libraries, and so on. Virtual Private Servers share one *Hardware Node* and one OS kernel. However, they are isolated from each other. Virtual Private Server is a kind of ‘sandbox’ for processes and users. Virtual Private Server 0 is used to designate the Hardware Node itself.

Virtuozzo is a complete server automation and virtualization solution allowing you to create multiple isolated *Virtual Private Servers* on a single physical server to share hardware, licenses, and management effort with maximum efficiency.

Virtuozzo Control Center is a tool designed for managing a particular *Hardware Node* and all *VPSs* residing on it with the help of a standard Web browser on any platform.

Virtuozzo File System (VZFS) is a virtual file system for mounting to *VPS* private areas. *VZFS* symlinks are seen as real files inside *VPSs*.

Virtuozzo license is a special license that you should load to the *Hardware Node* to be able to start using *Virtuozzo*. Every *Hardware Node* shall have its own *Virtuozzo* license file.

Virtuozzo Management Console is a *Virtuozzo* management and monitoring tool with graphical user interface. It uses *VZagent Protocol* to control *Hardware Nodes* and their *VPSs*. *VZMC* is cross-platform and runs on both Microsoft Windows and Linux workstations.

Virtuozzo Power Panels is a means for administering personal Virtual Private Servers with the help of a standard Web browser (Internet Explorer, Mozilla, etc.) on any platform.

VPS is an abbreviation of *Virtual Private Server*.

VPS 0 is used to designate a *Hardware Node* where *Virtuozzo* is installed.

VZagent (or *VZagent Protocol*) is an XML-based protocol used to monitor and manage a *Hardware Node*. The *vzagent* software implements this protocol and is a backend for the *Virtuozzo Management Console*.

vzagent0 is the user who has a full administrative access to the *Service VPS*. You will need to provide this user name and password when connecting to your *Hardware Node* by means of *VZMC* and *VZCC*.

VZCC is an abbreviation of *Virtuozzo Control Center*.

VZCC license is a license needed to activate *VZCC*. You should enter it the first time you log in to *VZCC*.

VZFS is an abbreviation of *Virtuozzo File System*.

VZMC is an abbreviation of *Virtuozzo Management Console*.

VZMC license is a license needed to activate *VZMC*. You should enter it the first time you launch *VZMC*.

VZPP is an abbreviation of *Virtuozzo Power Panels*.

Index

A

- About This Guide • 10
- About Virtuozzo Software • 15
- Accessing Devices From Inside Virtual Private Server • 250
- Action Scripts • 23, 56, 71
- Adding Application EZ Templates to VPS • 104
- Adding New Page • 203
- Adding Nodes to Cluster • 271
- Adding Skins to VZCC/VZPP • 208
- Adding Templates and Template Updates to VPSs • 81
- Adding VZCC/VZPP Links • 194
- Adjusting Colors and Styles • 160
- Adjusting Periodicity of Refreshing Information • 158
- Adjusting Representation Scale • 159
- Adjusting Template Configuration Files • 286
- Adjusting Traffic Representations • 198
- Administrator
 - Hardware Node • 24, 28, 123, 181, 193, 197
 - Plesk • 216
 - Service VPS • 197
 - system • 110
 - VPS • 74, 197, 199, 203
- Advanced Configuration • 184
- Advanced Tasks • 220
- Alerts • 165, 166
- Applications • 15, 18, 20, 26, 75, 78, 124, 127, 140, 143, 228, 259, 261, 263
- Assigning Access Rights to VZCC/VZPP User Groups • 266
- Assigning Default Backup Node • 60
- Associating VPS Files With Application Templates • 127
- Available Capabilities for Virtual Private Server • 222

B

- Backing Up and Restoring Caches • 128
- Backing Up and Restoring Virtual Private Servers • 56
- Backing Up Group of VPSs • 62
- Backing Up Single VPS • 60

Backup

- configuration file • 23, 57
- copy • 71
- directory • 23
- full • 56, 57
- incremental • 56, 57
- Node • 56, 62, 299, 300
- searching • 66
- timeout • 299
- utility • 57
- VPS • 28, 56, 62, 193, 197, 299, 300

Basic Configuration • 183

Basics of Virtuozzo Technology • 20

Before You Begin • 32

Building and Testing OS Template • 91

C

Capabilities Defined by POSIX Draft • 222

Changing Services Mode • 176

Changing System Time From VPS • 248

Checking for Conflicts Among Nodes • 275

Checking Quota Status • 123

Checking Virtual Private Server • 290

Choosing OS EZ Template • 38, 100

Choosing OS Template • 35

Choosing Virtual Private Server ID • 33

Cleaning Up Virtual Private Servers • 124

Cluster

- checking • 275
- configuring • 272, 273
- creating • 271
- destroying • 271
- network settings • 274
- overview • 270

Common Tasks • 194

Configuration Files

- backup • 23
- creating • 230, 257
- editing • 231
- global • 23, 51, 110, 112, 113, 129, 137, 183, 303
- Linux distribution • 224, 233, 245, 257
- managing • 146
- name-based hosting • 183
- template • 87, 286
- VPS • 23, 51, 71, 110, 112, 113, 129, 137, 146, 183, 224, 229, 233, 257
- vzagent • 197
- VZCC/VZPP • 193
- VZMC • 161
- Configuring Capabilities • 220
- Configuring Firewall • 282
- Configuring IP Addresses Pool • 272
- Configuring Network Bandwidth Management for VPS • 137
- Configuring Network Classes • 132
- Configuring Virtual Private Server • 44
- Configuring Virtual Private Server IDs Pool • 273
- Corrupted Pseudographics in Virtuozzo Utilities • 301
- Creating • 210
- Creating and Installing Application Templates • 78
- Creating and Installing Template Updates • 79
- Creating Cluster • 271
- Creating Configuration File for New Linux Distribution • 257
- Creating New Virtual Private Server • 31
- Creating OS Template • 85
- Creating OS Template in VZMC • 92
- Creating OS Template With vzsnapshot • 84
- Creating Virtual Private Server • 37, 40, 102
- Creating Virtual Private Servers in VZMC • 41
- Creating VPS Configuration File • 230
- Creating VPS on Basis of OS EZ Template • 37, 99
- Creating VPS on Basis of OS Standard Template • 34
- Creating VZFS Symlinks Inside VPS • 221
- Customizing Individual Pages • 201
- Customizing VPS Reinstallation • 69
- Customizing VZCC/VZPP • 192
- Customizing VZCC/VZPP for Using in Demo Mode • 212
- Customizing VZCC/VZPP Layout • 208
- Customizing VZCC/VZPP Menu • 206

D

- Defining Global Protocol Parameters • 185

- Defining VZCC/VZPP User Groups • 266
- Defining Window Manager to Run X Applications • 263
- Deleting Virtual Private Server • 71
- Detaching Virtual Private Server From Hardware Node Cache Directory • 128
- Determining VPS Identifier by Process ID • 177
- Differences Between OS and Application Templates • 23
- Disabling VPS • 73
- Disk Quota Parameters • 112
- Distinctive Features of Virtuozzo • 16
- DNS server • 37, 181, 235, 274
- Documentation Conventions • 13

E

- Editing VPS Configuration File • 231
- Enabling VPN for VPS • 253
- Establishing Secure Channel to SWsoft Support • 305
- Example • 186
- Extraneous Backups Visible to VPS in VZPP • 300
- EZ Template
 - adding to VPS • 93, 98, 104
 - application • 95, 98, 103, 104, 108
 - area • 95
 - caching • 93, 98
 - concepts • 93
 - directory structure • 95
 - installing • 98, 103
 - lifecycle • 98
 - listing • 103, 107
 - OS • 31, 38, 41, 93, 98, 107, 108
 - removing • 98, 109
 - repository • 93, 98
 - scripts • 95
 - sets • 93, 95
 - updating • 93, 98, 108
- EZ Template Directory Structure • 95
- EZ Template Lifecycle • 98

F

- Failure to Access VPS From Network • 298
- Failure to Back Up VPS in VZMC • 299
- Failure to Create VPS • 296
- Failure to Display List of VPS Backups • 299
- Failure to Log In to VPS • 298
- Failure to Run vgscan Utility • 301
- Failure to Start iptables Modules After Physical Server Migration • 300
- Failure to Start VPS • 297
- Feedback • 14

Files • 288
 Finding Kernel Function That Caused D
 Process State • 296
 Firewall • 17, 74, 282

G

General Considerations • 292
 General Conventions • 14
 Getting Assistance With VirtuoZZo Installation
 • 302
 Getting Technical Support • 302
 Glossary • 306

H

Hardware Node Availability Considerations •
 30
 Hardware Node Main Window • 27
 Highlighting Counter • 160
 HN • See Hardware
 Host OS • 17, 171, 229, 280, 306
 Hostname
 Hardware Node • 235
 proxy server • 303
 VPS • 28, 41, 66, 154, 180, 210, 229, 235,
 261
 HSPcomplete • 16, 306
 HTML • See Hyper Text Markup Language
 HTTP • See Hyper Text Transfer Protocol
 Hyper Text Markup Language • 199, 201, 206
 Hyper Text Transfer Protocol • 181, 186, 199,
 303

I

Importing Plesk Language Packs Into VZPP •
 219
 Installing Application EZ Templates on
 Hardware Node • 103
 Internet Explorer • 28
 IP Address
 Hardware Node • 27, 180
 mail relay server • 165
 peer-to-peer • 230
 physical server • 233, 250
 proxy server • 303
 Service VPS • 201
 VPS • 17, 20, 32, 37, 49, 154, 210, 224,
 257, 261, 270
 iptables • 253, 254, 255

K

Keeping Your VirtuoZZo System Up-to-Date •
 189
 Kernel

 2.4 • 226
 2.6 • 54

Kernel Troubleshooting • 294

L

Learning Current Name-Based Hosting Status
 • 188
 License
 VirtuoZZo • 24, 271, 292
 VZCC • 28
 Linux-Specific Capabilities • 223
 Listing EZ Templates • 107
 Listing Templates • 77
 Listing Virtual Private Servers • 49
 Loading iptables Modules • 253
 Loading iptables Modules to Hardware Node •
 254
 Loading iptables Modules to Particular VPSs •
 255
 Logs • 27, 157, 212, 285, 303

M

MAC Address • 228, 306
 Main Operations on Services and Processes •
 169
 Main Principles of VirtuoZZo Operation • 20
 Managing Backup Node • 65
 Managing Backups in VZMC • 59
 Managing CPU Share • 129
 Managing Demo Mode for VZPP With Plesk
 Integrated • 216
 Managing Disk Quotas • 111
 Managing EZ Templates • 93
 Managing Files • 288
 Managing Graphical Applications Inside VPS
 • 259
 Managing Mount Points • 284
 Managing Network Accounting and
 Bandwidth • 131
 Managing Node Clusters • 270
 Managing Offline Services Configuration •
 276
 Managing Processes and Services • 170
 Managing Resources • 110
 Managing Search Domains • 289
 Managing Services and Processes • 167
 Managing System Parameters • 140
 Managing Templates • 75
 Managing Users and Groups • 280
 Managing Virtual Private Server Servers • 289
 Managing VPS Resources Configuration • 146
 Managing VZCC/VZPP Access Rights • 265
 Mastering VZMC • 270
 Migrating Physical Server to VPS • 224, 233

Migrating Physical Server to VPS in
 Command Line • 229
 Migrating Physical Server to VPS in VZMC •
 235
 Migrating Virtual Private Server • 51
 Migrating VPS to Physical Server • 244, 246
 Migration
 physical server to VPS • 224, 229, 233,
 235, 300
 VPS to physical server • 244, 245, 246, 250
 VPS to VPS • 51
 Migration Overview • 224
 Migration Requirements • 226, 246
 Migration Restrictions • 228
 Migration Steps • 224, 245
 Miscellaneous Problems • 301
 Modifying Existing Page • 201
 Monitoring Memory Consumption • 145
 Monitoring Processes in Real Time • 173
 Monitoring Resources in Text Console • 154
 Monitoring Resources in VZMC • 156
 Monitoring System Resources Consumption •
 143
 Moving Network Adapter to Virtual Private
 Server • 252
 Moving VPS Files to Cache Area on Hardware
 Node • 126
 Mozilla • 28, 264

N

Name-Based Hosting
 checking status • 188
 configuration • 183, 184
 example • 186
 overview • 181
 Name-Based Hosting for Virtuozzo • 180
 Name-Based Hosting Overview • 181
 Network
 adapter • 135, 137, 228, 230, 252, 253
 bandwidth • 135, 137, 198
 classes • 132, 135, 137, 199
 configuration • 278
 connection • 228, 246, 249
 interface • 183, 231
 local • 181
 parameters • 45, 110, 131, 235, 274, 303
 public • 261
 services • 180, 183
 traffic • 131, 134, 137, 198
 virtualization • 17
 VZMC • 26
 Network Traffic Parameters • 131
 Network Virtualization • 17
 Node

Backup • 56, 57, 60, 64, 66
 Client • 271
 Destination • 51, 62
 Hardware • 11, 19, 21, 26, 27, 28, 30, 32,
 47, 51, 77, 112, 129, 132, 140, 146, 171,
 181, 197, 206, 210, 224, 231, 245, 249,
 254, 270, 272, 278
 Master • 271, 272, 273
 Source • 60, 62, 66, 67
 Target • 51

O

Obtaining Hardware Node ID From Inside
 Virtual Private Server • 249
 Offline Management • 276
 Operations on Virtual Private Servers • 31
 Organization of This Guide • 12
 OS Template Updates vs Application
 Templates • 83
 OS Virtualization • 17
 Overview • 93, 193, 259

P

Password
 root • 37, 46, 199, 233, 246, 261
 setting • 46
 VPS user • 30
 Plesk • 29, 146, 208, 216, 219, 280
 Pool • 23, 135, 271, 272
 Preface • 10
 Preparing and Sending Questions to Technical
 Support • 302
 Preparing OS EZ Template for VPS Creation •
 98
 Preparing Packages • 87
 Preparing Virtual Private Server for Building
 OS Templates • 86
 Preparing VPS Configuration File • 229
 Problems With Physical Server Migration •
 300
 Problems With VPS Management • 296
 Problems With VPS Operation • 300
 Processes
 monitoring in real time • 173
 overview • 167, 168, 169
 PID • 177
 viewing • 171
 Processor
 32-bit • 51, 56, 140
 64-bit • 51, 56, 140

Q

QoS • See Quality of Service
 Quality of Service • 19, 110, 165, 210, 235

R

RAID • See Redundant Array of Inexpensive Drives

RAM • See memory

Real-Time Monitoring in Virtuozzo • 153

Rebooting Virtual Private Server • 258

Red Hat Package Manager • 18, 21, 84, 87

Redundant Array of Inexpensive Drives • 30

Reinstalling Virtual Private Server • 67

Removing Application EZ Template From VPS • 109

Removing Templates and Template Updates From VPSs • 82

Replaying Information From Logs • 161

Resource Management • 19

Resources

- configuration • 146, 148, 149, 151
- CPU • 17, 19, 110, 129, 235
- disk space • 19, 110, 111, 112, 115, 118, 120, 123, 235
- memory • 19, 110, 143, 145, 235
- monitoring • 154, 156, 157, 158, 159, 160, 161, 164, 165, 166
- network • 110, 131, 132, 134, 135, 137
- overview • 110
- sharing • 21
- system • 110, 143, 171, 235

Restoring Group of VPSs • 64

Restoring Single VPS • 62

root

- account • 193, 201
- operating system • 20, 84
- partition • 20, 85
- password • 29, 37, 46, 199
- user • 194, 233, 246

Routing

- rules • 16
- table • 17, 282

RPM • See Red Hat Package Manager

Running Backup Utilities on Backup Node • 57

Running Commands in Virtual Private Server • 74

Running Graphical Applications in X Windows • 259

Running Graphical Applications via VNC • 264

S

Saving and Decoding Kernel Fault (OOPS) • 295

Saving Counters Configuration • 161

Scaling Virtual Private Server Configuration • 149

Scripts • 17, 18, 23, 47, 56, 71, 143, 224, 231, 257, 263, 290, 292

Search Domain • 289

Searching for VPS Backups • 66

Secure Shell • 30, 74, 193, 224, 233, 246, 261, 306

Service Level Agreement • 19

Service VPS • 26, 29, 47, 171, 193, 194, 197, 198, 199, 201, 203, 206, 208, 210, 216, 226, 271, 276

Services

- changing mode • 176
- offline • 276
- overview • 168, 169
- restarting • 178
- starting/stopping • 178
- viewing • 171
- xinetd-dependent • 167, 169, 170, 176, 178

Setting Auto Logout Time • 196

Setting Maximal Backups Number • 197

Setting Network Parameters • 45

Setting root Password for VPS • 46

Setting Startup Parameters • 44

Setting Up Per-VPS Disk Quota Parameters • 115

Setting Up Second-Level Disk Quota Parameters • 120

Setting Up SMTP Disk Quota • 187

Setting Up VPS Permissions • 186

Setting Up VZCC/VZPP Demo Mode for Potential Customers • 210

Sharing File System Among Virtual Private Servers • 256

Shell Prompts in Command Examples • 14

SLA • See Service Level Agreement

Specifying Cluster Network Settings • 274

Splitting Hardware Node Into Equal Pieces • 148

SSH • See Secure Shell

Standard Migration • 51

Starting, Stopping, and Restarting Services • 178

Starting, Stopping, Restarting, and Querying Status of Virtual Private Server • 47

Submitting Problem Report to Technical Support • 303

Subscribing for Alerts • 165

Subscribing for VZMC Alerts • 165

Subscribing for vzrmond Alerts • 166

Support • 302, 303, 305

Swap • 27, 140, 145, 154, 278

Symlinks

creating • 221
overview • 21, 37

T

Tarball • 23, 57
TCP • 110, 140, 183, 186, 210, 216, 259, 306
Telnet • 45, 282
Template
 adding • 76, 81
 alert • 165
 application • 18, 21, 23, 35, 75, 77, 78, 81, 82, 83, 87, 126, 127
 area • 124
 base • 78, 79
 caching • 23, 76, 87
 choosing • 37
 configuration file • 85, 87, 91, 286
 creating • 76, 78, 84, 85
 directory • 21, 23, 78, 91, 124
 files • 21, 124
 HTML • 199
 installing • 21, 76, 78
 lifecycle • 76
 listing • 77
 management utilities • 77, 78, 79, 82, 84, 91
 OS (operating system) • 18, 21, 23, 35, 75, 77, 83, 84, 85, 86, 87, 91, 92, 224, 233
 overview • 18, 21, 23
 removing • 76, 82
 updates • 21, 75, 79, 81, 82, 83, 127
Template Lifecycle • 76
Templates • 18
Timeout When Accessing Remote Hosts • 300
Troubleshooting • 291
Turning On and Off Network Bandwidth Management • 135
Turning On and Off Per-VPS Disk Quotas • 113
Turning On and Off Second-Level Quotas for Virtual Private Server • 118
Typographical Conventions • 13

U

UBC • See User Beancounters
UDP • See User Data Protocol
Understanding EZ Templates • 93
Understanding Licensing • 24
Understanding Templates • 21
Update

 adding to VPS • 81
 creating • 79
 installing • 79
 operating system • 83
 removing • 82
 template • 21, 75, 79, 81, 83, 124
Updating EZ Templates in VPS • 108
Updating Host OS Software • 189
Updating Software Inside Virtual Private Servers • 191
Updating Virtuozzo Software • 191
User
 demo • 210, 212
 level • 110
 managing • 280
 quota • 112, 115, 123, 224, 233, 244
 root • 194, 233, 246, 261
 Service VPS • 206
 Virtuozzo • 305
 VPS • 17, 21, 30, 37, 74
 VZCC • 28, 266
 VZPP • 29, 266
User Beancounters • 140, 143, 235, 306
User Data Protocol • 140, 282
Using ALT+SYSRQ Keyboard Sequences • 294
Using Charts Representation • 157
Using Table Representation • 164
Using up2date • 190
Using X Windows to Run Graphical Applications • 261
Using yast • 190
Using yum • 190
Utilities
 backup management • 23, 57
 EZ template management utilities • 98, 103, 104, 107, 108, 109
 license management • 24
 migration management utilities • 51, 224, 229, 233, 245, 246
 resources management utilities • 113, 115, 118, 120, 123, 126, 127, 128, 129, 134, 135, 137
 template management • 77, 78, 79, 81, 82, 84
 VPS management • 37, 44, 45, 46, 47, 49, 67, 71, 74

V

Validating Virtual Private Server Configuration • 151
venet • 228, 230, 306
Viewing Active Processes and Services • 171
Viewing Network Traffic Statistics • 134

- Viewing Summary Pages • 278
- Viewing System and Virtuozone Logs • 285
- Virtual Network Computing • 259, 264
- Virtual Private Network • 253, 305
- Virtual Private Server
 - accessing • 74
 - administrator • 29, 194, 197, 210
 - backing up • 56, 57, 60, 62, 197
 - checking • 290
 - checking status • 47
 - cleaning up • 124
 - configuration file • 23, 129, 146, 149, 151, 183, 229, 231, 246, 257
 - configuring • 44, 45, 46
 - copy-on-write area • 21
 - CPU share • 129
 - creating • 23, 31, 32, 33, 35, 37, 41
 - destroying • 71
 - disk quota • 112, 113, 120, 123, 124
 - files • 124, 126, 127, 288
 - hostname • 41, 45
 - IDs pool • 273
 - IP address • 41, 45, 183, 216, 270, 271
 - listing • 49
 - migrating • 51, 224, 233, 235, 244, 246
 - mount point • 284
 - name-based • 180
 - network parameters • 131, 132, 134, 135, 137
 - permissions • 186
 - private area • 21, 23
 - rebooting • 258
 - reinstalling • 67, 69
 - restarting • 47
 - restoring • 56, 57, 62
 - servers • 289
 - starting/stopping • 47
 - symlinks • 21, 23
 - system parameters • 140, 143, 145
 - understanding concepts • 16, 17
 - updating • 108
 - user • 74, 197, 280
- Virtualization
 - network • 17
 - operating system • 17
- Virtuozone • 17
 - 32-bit version • 51
 - 64-bit version • 51
 - applications • 16
 - capabilities • 15
 - cluster • 270
 - configuration file • 23, 37, 51, 183, 285
 - file system • 18
 - installing • 31, 302
 - layer • 20
 - license • 24
 - logs • 285
 - name-based hosting • 180
 - network • 17
 - Quality of Service • 110
 - support tunnel • 305
 - technology • 15, 16, 17, 18, 19, 20
 - templates • 18
 - tools • 10, 24, 28, 29, 169, 305
 - utilities • 23
- Virtuozone • 24
- Virtuozone Agent • 26, 197, 199, 203, 208, 226, 271, 299, 306
- Virtuozone Applications • 16
- Virtuozone Configuration • 23
- Virtuozone Control Center • 10, 28, 192, 193, 194, 199, 201, 206, 208, 210, 224, 265, 266, 276
- Virtuozone File System • 18, 21, 37, 221, 228, 306
- Virtuozone File System (VZFS) • 18
- Virtuozone Management Console • 10, 21, 24, 25, 26, 27
- Virtuozone Philosophy • 15
- Virtuozone Power Panels • 10, 29, 169, 192, 199, 206, 208, 210, 219, 265, 266, 276
- VNC • See Virtual Network Computing
- VPN • See Virtual Private Network
- VZAgent • See Virtuozone Agent
- vzagent0 • 197, 306
- VZCC • See Virtuozone Control Center
- VZCC Overview • 28
- VZCC/VZPP Architecture • 199
- VZFS • See Virtuozone File System
- VZMC Network Architecture • 26
- VZMC Overview • 24
- VZMC Specific Features • 25
- VZPP • See Virtuozone Power Panels
- VZPP Overview • 29

W

- What are Disk Quotas? • 112
- What are Resource Control Parameters? • 110
- What Are Services and Processes • 168
- What is Virtuozone • 15

Who Should Read This Guide • 11

Writing Template Configuration File • 87

X

X Window System • 26, 259, 261, 263

XHTML • 201

XML • 26, 193, 194, 198, 203, 206

XSL • 193, 199, 201, 208

XSLT • 208

Z

Zero Downtime Migration • 15, 51, 54

Zero-Downtime Migration • 54